International Journal of Current Engineering and Technology
©2017 INPRESSCO®, All Rights Reserved

E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
Available at http://inpressco.com/category/ijcet

Research Article

An Approach to Detect Alteration in Text Document

Antima Singh[†*] and Sanjay Kumar Yadav[†]

[‡†]CS & IT, SAM Higginbottom University of Agriculture, Technology & Sciences, Allahabad, India

Accepted 01 April 2017, Available online 05 April 2017, **Vol.7, No.2 (April 2017)**

Abstract

In almost all watermarking based documents authentication systems, the documents are considered as binary images and then, the watermark is embedded using some image watermarking algorithms. Important text documents files are saved in a disk space and other type of storage systems. The data moves from one network to another network. Therefore, while the data travelling in this kind of network privacy and security management is an essential concern of security. If any people did tamper attack in text document file, Tamper attack can't be found easily in the text file. The fragile watermarking Algorithms are usually used in building text authentication system for avoiding the criticalness the data using fragile watermarking Algorithm. This paper proposes a secure fragile watermarking Algorithm. This is an extension of an existing data hiding scheme with binary images. The given concept is implemented using MATLAB, and the performance of Algorithm is evaluated using the different performance parameters that are compression, encryption, detection, decryption, confidentiality and Integrity of data using secret key Authentication technique. The computed result shows the effectiveness and efficiency of the methodology for text document authentication that can be adoptable through a wide verity of application.

Keywords: *Text document authentication, Fragile watermarking technique, secret key, Gray scale Image.*

1. Introduction

In the recent years the text document venerability attacks increasing easily by the unintended users. In the same manners the cases of security issues and frauds are also increasing. The main reason behind this is unsecure data transfer in the network. To prevent user data over the un-trusted network most of the time cryptographic approaches are utilized, but traditionally available algorithms are not much suitable in the new generation computing technology. Therefore, new cryptographic scheme is required to provide by which the data security becomes robust. In recent time, cryptography has turned into a battleground of some of the world's best mathematician and computer scientists the ability to securely store and transfer sensitive information has proved a critical factor of business. In order to find more effective, robust and efficient cryptographic technique, a large number of cryptographic approaches are available. In this proposed work specific data oriented cryptographic techniques are investigated, namely image cryptography.

There are several ways of classifying cryptographic algorithms. In this paper, discussed these type of cryptography only for prevent the data with the help of different type of cryptography Algorithms.

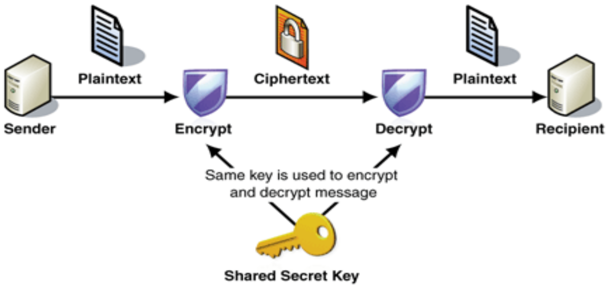


Figure1: Overview of cryptography using symmetric key

This is discussed in it. Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption; also called symmetric encryption. Primarily used for privacy and confidentiality. Public Key Cryptography (PKC): Uses one key for encryption and another for decryption; also called asymmetric encryption. Primarily used for authentication, non-repudiation, and key exchange. Hash Functions: Uses a mathematical transformation to irreversibly encrypt information, providing a digital fingerprint. Primarily used for message integrity.

1.1 Text cryptography

In text cryptography, encryption is the process of transforming information using algorithm, which

*Corresponding author: **Antima Singh**

means that original text transform into cipher text, cipher text known as encrypted or encoded information that is unreadable by a human or computer without the proper cipher to decrypt it. Decryption is the inverse of encryption.

1.2 Text file compression

In text cryptography, firstly plain text converts into ASCII form, which will be decimal number. All decimal numbers are converted into eight-bit binary form. After this, have huge amount of binary stream, which is take more memory, for overcome this problem using compression technique. In compression technique, run length encoding technique applying on binary bit stream. After this, it will be represented as compressed text file which will be no more readable.

1.3 Authentication bit generation and text file to image conversion

Once we get the compressed text file it will be embedded with authentication bit in order to make self-authenticating text files. These authentication bits are nothing but fragile watermark, which will be destroyed if any alteration is done to text file.

2. Background

The sudden increase in watermarking is most likely due to the increase in concern over copyright protection of content and content authentication of digital media. The internet is an excellent distribution system for digital media because it is inexpensive and delivery is almost instantaneous. However, the owners of the content also see a high risk of piracy. The risk of piracy is exacerbated by the proliferation of high-capacity digital recording devices. Using these recording devices and using the internet for distribution pirates can easily record, alter and redistribute the copyright protected material without appropriate compensation being paid to the actual copyright owners.

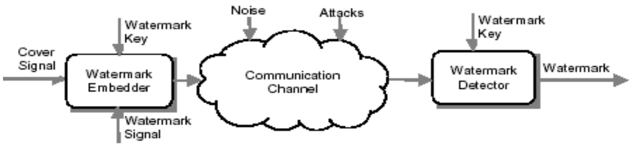


Figure2: Digital watermarking system

2.1Watermark Introduction

A watermark is a more or less transparent image or text that has been applied to a piece of paper, another image to either protect the original image or text. Usually watermarking is divided into visible and invisible, fragile and robust, spatial and frequency.

Watermarking in actual fact is an important content in information hiding, however it emphases the little tags consisting of random binary numbers. Watermarking is designed to protect copyright by embedding secret to the host media, according to the differences of functionalities and appearance of watermarking; watermarking has been grouped into many categories. This is showing in (Figure3).

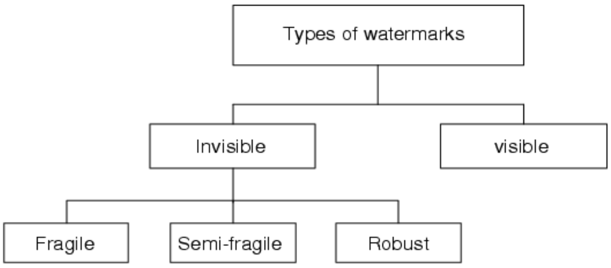


Figure3: Types of Watermarks

2.2 Visible Watermark

Visible watermark such as logo can be seen on the visual media such as images, photos and videos. Visible watermarking used to indicate ownership originals. The typical example is that television station marks their logos at the corner of the screen while plying TV drams, news, programs.

2.3 Invisible watermark

Invisible watermarking is the validate way to identify original author, ownership, distributors; Invisible watermarking cannot be visually touched. There are some invisible watermarks:

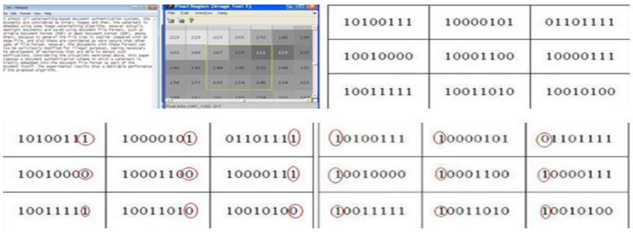
2.3.1 Robust Watermarking: Robust watermarking refers to the tampering to the watermarks media, the watermarks can be restored from the destroyed media.

2.3.2 Fragile watermarking: Fragile watermarking always is suitable for detection of the minor changes in digital media; this is very helpful in detecting the integration of the media.

2.3.3 Semi-Fragile Watermarking: Semi-fragile watermarks are more robust than fragile watermark and less sensitive to classical user modification.

2.4 Authentication 8-bit Representation and text file to image conversion

Once we get the compressed text file it will be embedded with authentication bit in order to make self-authenticating text files. These authentication bits are nothing but fragile watermark, which will be destroyed if any alteration is done to text file. In fragile watermarking, a digital watermark is called fragile if it fails to be detectable after the slightest modification. Fragile watermarks are commonly used for tamper detection (integrity proof). Digitally, an image is represented in terms of pixels. These pixels can be expressed further in terms of bits. Consider the text image and the pixel representation of the image. Consider the pixels that are bounded within the yellow line. The binary formats for those values are (8-bit representation). The binary format for the pixel value 167 is 10100111similarly, for 144 it is 10010000.This 8-bit image is composed of eight 1-bit planes. Plane 1 contains the lowest order bit of all the pixels in the image. And plane 8 contains the highest order bit of all the pixels in the image.



10100111	10000101	01101111
10010000	10001100	10000111
10011111	10011010	10010100

10100111	10000101	01101111	10100111	10000101	01101111
10010000	10001100	10000111	10010000	10001100	10000111
10011111	10011010	10010100	10011111	10011010	10010100

Figure 4: Example of 8-bit Representation with lowest bit and highest bit

3. Proposed Work

3.1 The Proposed Watermarking Scheme

In this section, it introduces in detail our proposed watermarking scheme for exact authentication of text documents. It defines how to keep text documents safe by using fragile watermarking. Any changing will happen in the text document then the detection process will find out something change in text documents. In this section it will present thought of blocking text documents, mathematical model and definitions, and steps of the watermarking scheme. Firstly, plain text converts into ASCII form, which will be decimal number. All decimal numbers are converted into eight-bit binary form. After this, have huge amount of binary stream, which is take more memory, for overcome this problem using compression technique. In compression technique, using self-embedding technique which name is run length encoding technique. Run length encoding technique applying on binary bit stream. After this, it will be represented as compressed text file which will be no more readable. Once we get the compressed text file it will be embedded with authentication bit in order to make self-authenticating text files. These authentication bits are nothing but fragile watermark, which will be destroyed if any alteration is done to text file.

3.2 Introduction to MATLAB R2010a

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language developed by Math Works. The Proposed scheme has been implemented on MATLAB R2010a. It is a high level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB R2010a, it can solve technical computing problems faster than with traditional programming languages such as c, c++. One can use MATLAB R2010a in a wide range of application, including signal and image processing, communications, control design, test and measurement, and computational biology. Add-on toolboxes extend the MATLAB environment to solve particular classes of problems in these application areas.

3.3 Flowchart of Proposed Approach

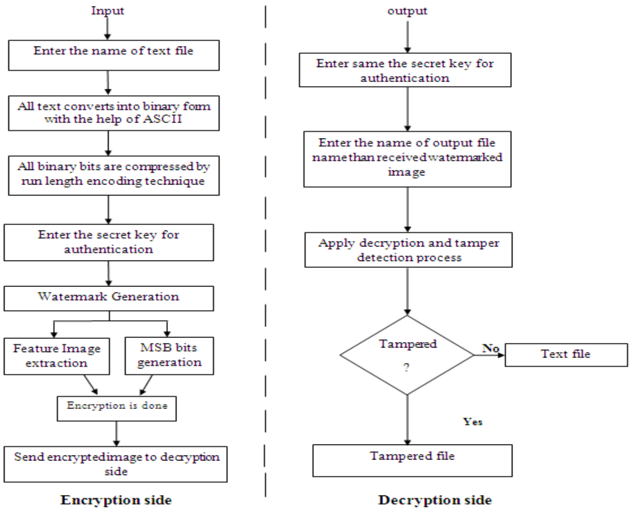

Input
Enter the name of text file
All text converts into binary form with the help of ASCII
All binary bits are compressed by run length encoding technique
Enter the secret key for authentication
Watermark Generation
Feature Image extraction
MSB bits generation
Encryption is done
Send encryptedimage to decryption side
Encryption side
output
Enter same the secret key for authentication
Enter the name of output file namethan receivedwatermarked image
Apply decryption and tamper detection process
Tampered ?
No
Text file
Yes
Tampered file
Decryption side

Figure 5: Flowchart

3.4 Steps of Proposed Approach

As we know that text file is nothing but the combination of various alphanumeric characters which includes number, characters and special symbols. All given four steps are hierarchal. It means output of previous step will be the input of next steps. Proposed approach consists of four steps as shown in (figure6).

3.4.1 Text files compression using run length encoding: Take a text files as an input and read it's all characters' row wise. Convert all characters into its ASCII form, which will be decimal number. All decimal numbers are converted into eight-bit binary form. Apply run length encoding on binary bit stream. Output of step above will be represented as compressed text file which will be no more readable.

3.4.2 Authentication bit generation and text file to image conversion: Take the compressed text file as an input. Convert the compressed text file into binary vector. Create clusters of five bits for all bits of text file binary vector. Generate three authentication bits for each cluster of five bits by following way.

For 1st bit generation- Take bit wise XOR of five bits of each cluster and calculate the modulo2 of the sum of them.

$$1stbit = (\sum_{i=1}^{4} b_i \oplus b_{i+1}) mod\ 2 \qquad (1)$$

Where bi represents the ith bit of vector

For 2nd bit generation- Using a secret key, generate a random matrix Rm of same size r x c as image whose values ranges from 0 to 31. Do bit wise XOR between corresponding cluster's bits and bits of Rm.

$$2ndbit = \left(\sum_{i=1}^{5} b_i(Im) \oplus b_i(Rm)\right) mod\ 2 \qquad (2)$$

For 3rd bit generation- Calculate the decimal value of five-bit cluster. Calculate the complement of each decimal value of cluster and take the bit wise XOR with its original value.

$$3rdbit = \left(\sum_{i=1}^{5} b_i(Im) \oplus b_i(31 - Im)\right) mod\ 2 \qquad (3)$$

Append all generated three bits to five bits of its corresponding cluster as shown in figure. Now takes the decimal value of each eight-bit binary cluster. The resultant matrix will be represented as image.

3.4.3 Tamper detection: In tamper detection, take the altered text file image as an input. Take a tamper localization matrix with all initially assigned values as 0. Extract the last three bits of all pixels of text file image. Recalculate the three bits for all clusters using eq. 1, 2 and 3 Compare the extracted and recalculated three bits for all corresponding pixels. If any mismatch is found, then marks the corresponding location in tamper localization matrix.

3.4.4 Text file decompression: Extract five MSBs of each pixels of image text file and make a vector. Now creates clusters of eight bits from given vector. All eight bit clusters are converted into decimal format. Now again creates clusters of eight bits from given vector.

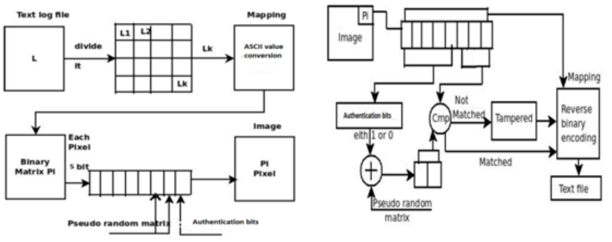
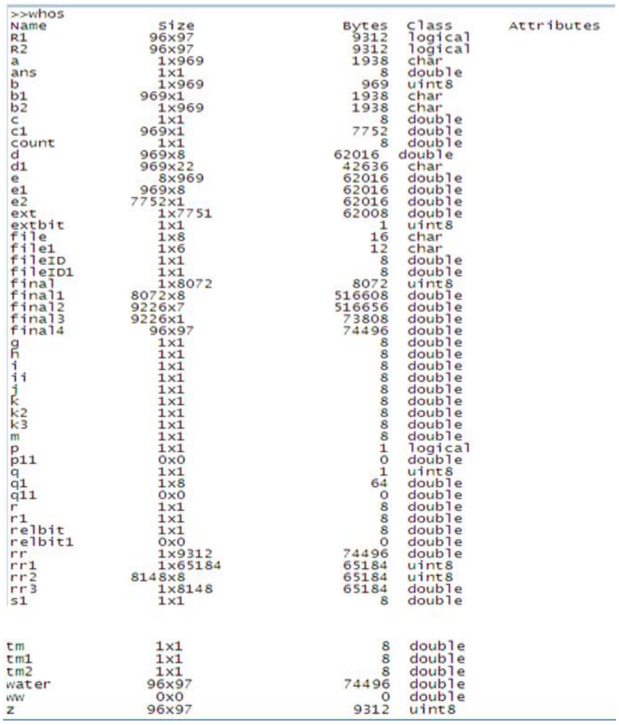


Figure 6: Authentication bit generation and text files to image conversion and tamper localization procedure

3.5 Variable information

(Table1) shows the detail of variable which have used in the coding of Proposed Watermarking Scheme. When whos command is run in the mat lab then it will show those entire variable which were used in code. It has some attributes; their names are NAME, SIZE, BYTES, and CLASS, ATTRIBUTES defined respectively. Name: Name of the variable. Size: Dimensions of the variable array. Bytes: Number of bytes allocated for the variable array. Class: Class of variable. If the variable has no value, class is '(unassigned)'

Table 1: Variable Detail



```
>>whos
Name              Size                  Bytes   Class      Attributes
R1               96x97                   9312   logical
R2               96x97                   9312   logical
a                 1x969                  1938   char
ans               1x1                       8   double
b                 1x969                   969   uint8
b1              969x1                    1938   char
b2                1x969                  1938   char
c                 1x1                       8   double
c1              969x1                    7752   double
count             1x1                       8   double
d               969x8                   62016  double
d1              969x22                  42636   char
e                 8x969                 62016   double
e1              969x8                   62016   double
e2             7752x1                   62016   double
ext               1x7751                62008   double
extbit            1x1                       1   uint8
file              1x8                      16   char
file1             1x6                      12   char
fileID            1x1                       8   double
fileID1           1x1                       8   double
final             1x8072                 8072   uint8
final1         8072x8                  516608   double
final2         9226x7                  516656   double
final3         9226x1                   73808   double
final4           96x97                  74496   double
g                 1x1                       8   double
h                 1x1                       8   double
i                 1x1                       8   double
ii                1x1                       8   double
j                 1x1                       8   double
k                 1x1                       8   double
k2                1x1                       8   double
k3                1x1                       8   double
m                 1x1                       8   double
p                 1x1                       1   logical
p11               0x0                       0   double
q                 1x1                       1   uint8
q1                1x8                      64   double
q11               0x0                       0   double
r                 1x1                       8   double
r1                1x1                       8   double
relbit            1x1                       8   double
relbit1           0x0                       0   double
rr                1x9312                74496   double
rr1               1x65184               65184   uint8
rr2            8148x8                   65184   uint8
rr3               1x8148                65184   double
s1                1x1                       8   double


tm                1x1                       8   double
tm1               1x1                       8   double
tm2               1x1                       8   double
water            96x97                  74496   double
ww                0x0                       0   double
z                96x97                   9312   uint8
```

4. Result Analysis

According to our scheme we first take text files as an input to the algorithm as shown in figure (7.1example of text files). First of all text file is compressed by proposed compression technique. After compression it will be no more readable in nature. Now this compressed text file again embedded with some authentication bits in order to make its self-authenticable text files as shown in figure (7 encrypted image text file).

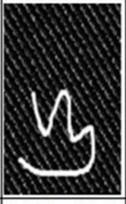
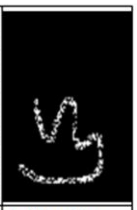
			
1. Example of text files	2. Encrypted image text file	3. Altered text files	4. Alteration detected text file.

Figure7: Result Analysis

Suppose an attacker has done some alteration in image text files like shown in figure (7.3Alteration text files). Here the white colored area shows the tampered portion in image (7.4Alteration detected text file) text file which was very difficult to spot in text file because text can easily be modified. After ensuring that image text file is altered somewhere, when we convert those image text file in to text file then it will not be similar like original one. Hence by proposed algorithm then one can easily decide that integrity of text file is compromised.

4.2 Computational Time of character

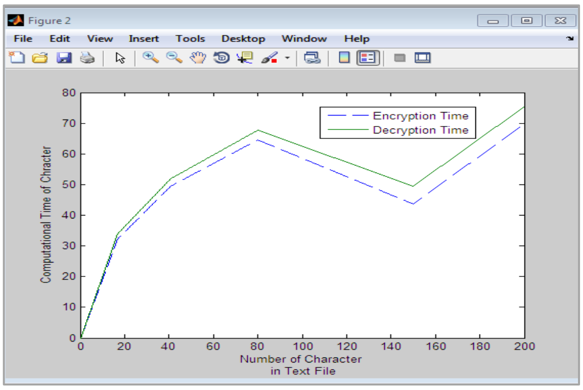


Figure8: Computational Time of character

(Figure 8) shows the computational time of character with encryption time and decryption time. It shows the how much take time at the side of encryption, whenever text file is encrypted and also shows the decryption time whenever text file is decrypted at the side of decryption. Each time vary the time of encryption time and decryption time, while it takes same file of text each time. In the figure (8) also shows the computational time of character goes down due to some processing files of encryption and decryption are already saved when some files are randomly run because of this it goes down. After this again it will go increase time of computational time of character.

Conclusion

1) The proposed study is intended to design a security algorithm against the previously available algorithm for text cryptography.
2) The previously available method having some limitations such as less cipher strength, limited for detection of text and the size of cipher text therefore a new fragile watermarking approach is required to design for finding the more appropriate and efficient solution.
3) The computational time of character with encryption time and decryption time of text is very less than others available method.
4) In order to find the targeted goal various similar research articles are studied and the solution is formulated.
5) This solution is based on the bit plane based lossless encryption methodology additionally the proposed technique consumes the run length encoding technique to improve the time complexity in place of previously available algorithm.
6) The previously available algorithm is basically an idea of consuming too long time for processing. In addition of that the gray scale image bit planes are modified for achieving high quality cipher in less time and memory consumption.
7) The implementation of the proposed methodology is provided using mat lab environment. The performance of the system is evaluated in terms of memory and time complexity.

References

Anbo Li, Bing-Xian Lin, Ying Chen (2008), *Study on copyright authentication of GIS vector data based on Zero-watermarking,* The International Archives of the Photogrammetric, Remote Sensing and Spatial Information Sciences. Vol. VII. Part B4, pp.1783-1786, 2008.

CHE Shengbing, MA Bin and CHE Zuguo (2008), *An Adaptive and Fragile Image Watermarking Algorithm Based on Composite Chaotic Iterative Dynamic System.* IEEE DOI 10.1109/IIH-MSP, 24.

C. Culnane, H. Treharne, and A.T.S. Ho (2008), *Authenticating Binary Text Documents Using a Localising OMAC Watermark Robust to Printing and Scanning.* Vol. VII. Part A4, pp.1783-1789, 2008.

Fridrich J *et.al* (2000) *New fragile authentication watermark for image.* ICIP'2000, Vancouver, Canada Vol. Viii. Part A4, pp.1783-1789(2000).

Gonzalez-Lee, M., Santiago-Avila, C., Nakano-Miyatake, M., & Perez-Meana, H. (2009). *Watermarking based document authentication in script format.* Proceedings of the 52th IEEE Midwest Symp. On Circuits and Systems, voll. 1, 837-841.

Jitendra Jain *et.al.* (2014) *Digital Image Watermarking Based on LSB for Gray Scale Image.* IJCSNS, VOL.14 No.6, June 2014.