Hiding Secret Message using Enhanced MSA Algorithm

Dr. M. Nandhini

Department of Computer Science, Pondicherry University.
mnandhini2005@yahoo.com

Abstract: The Secret Intelligence Agencies (SIA) is providing their secure services to our government with foreign intelligence The SIA has always used undercover agents to solve complex cases and dismantle criminal organizations. This work was conceptualizing as a solution for that process. For the secure communication, the advanced steganographics technique has been proposed In the basic approach of steganography, the cover file formats like images (BMP, JPG, etc), videos (MP4, etc) has been used. In our proposed approach, secret messages are hidden in encrypted form using enhanced MSA algorithm for non standard cover files such as .docx, .pptx, .xlsx, .pdf, .mdb, .exe, etc.,. To make the system fully secured, encrypt the secret message using enhanced MSA algorithm along with different randomization methods and then hide the encrypted message inside the cover file. To hide encrypted message, insert the 8-bits in two consecutive bytes of cover file in LSB, LSB+1, LSB+2, LSB+3 positions. The proposed system is more secured since enhanced MSA algorithm has been used with different randomization methods compare to existing work where, general MSA algorithm has been implemented.

***Keywords*— Steganography, Cryptography, Data encryption and decryption, Cover files, Secret message**

1. INTRODUCTION

The Secret Intelligence Agencies (SIA) is providing their secure services to our government with foreign intelligence. It operates under the formal direction of defense intelligence. The SIA has always used undercover agents to solve complex cases and dismantle criminal organizations. This project was conceptualizing as a solution for that process, so the SIA and their agents can communicate through this application for the exchange of evidences in a secure way. The SIA are needed to protect the citizens against clandestine and convert operations from the other countries and internal sources. They also act to secure a country's valuable information and prevent coups and the creation of instability within a country. Exchanging data over the internet is a critical issue due to security problems. Transmitting data from source to a destination is not easy task in the sense of security, and hiding data such as text documents into JPEG image to prevent these documents from attacks is well known problem in the area of data security.

To achieve this secure data transmission of evidences, the advanced steganographic Least Significant Bit (LSB) technique is proposed to hide data such as .doc file into .exe format and send the .exe file from source to any destination through internet, and then extract the hidden file by using a special application. To increase the security and the size of stored data, a new LSB technique with enhanced MSA algorithm is used. Instead of storing the data in every least significant bit of the binary values, this technique tries to use more than one bit in binary value in such a way that this change will not affect the host file. It uses the side information of neighboring bit values to estimate the number of bit which can be carried in the binary value of the host-file to hide the secret data.

Steganography is a technique for hiding the messages like text, audio, video, image in any cover file format. In earlier days, the people used only standard cover file such as image, text, etc., for hiding the secret information which does not include any non-standard cover file formats. In our proposed approach, secret messages are hidden in encrypted form using enhanced MSA algorithm for non standard cover files such as .docx, .pptx, .xlsx, .pdf, .mdb, .exe, etc.; however, the size of the hidden message must be very small in comparison to cover file.

To make the system fully secured first encrypt the secret message using enhanced MSA algorithm along with different randomization methods and then the encrypted message inside the cover file has hidden. To hide encrypted message, insert 8-bits in two consecutive bytes of cover file in LSB,

LSB+1, LSB+2 and LSB+3 positions The proposed system is more secured than the existing work due to the proposed enhanced MSA algorithm with different randomization methods.

2. LITERATURE SURVEY

In the existing works the data hiding has been achieved by using different data hiding concepts. In 2011, Joyshree Nath et al.[1] explained about hiding encrypted message in LSB and LSB+1 positions. It basically shows how one can hide information in encrypted form to any cover file such as .exe files, Microsoft office files, .dbf files, image files, audio files and video files. To make the system fully secured it first encrypt the secret message using MSA algorithm (Nath et al.[1]) and then hide the encrypted message inside the cover file. For hiding secret message it has changed both LSB and LSB+1 bits of each byte of the cover file. The MSA [1] algorithm introduced a new randomization method for generating the randomized key matrix to encrypt plain text file and to decrypt cipher text file. This method may most suitable for water marking.

Asoke Nath, Joyshree Nath (2011) described a steganography algorithm for secret message hiding [2]. In their work they tried to embed some secret message inside any cover file in encrypted form. It uses the standard steganographic method i.e. changing LSB bits of the cover file. For encryption method they use maximum encryption number=64, key_text and maximum randomization number=128. The merit of this method is that if change in key_text occurs little bit then the whole encryption and decryption process changes.

In 2011, Dripto Chatterjee et al. presented a DJSSA symmetric key algorithm for secure data hiding. It dealt with modified advanced symmetric key cryptographic method i.e. modified DJSSA algorithm [3] for multiple encryption and decryption of any file. This method was an extension of MSA algorithm proposed by Nath et al [1].

Dipti Kapoor Sarmah and Neha Bajpai (2010) described AES algorithm [4] for data hiding using cryptography and steganography. This algorithm uses Secret Key Cryptography (SKC), Public Key Cryptography (PKC), and Hash Functions for encryption. It is very difficult to detect hidden message in frequency domain and for this domain and it uses various transformations like DCT, FFT and Wavelets etc. In this work they are developing a system where develop a new technique in which cryptography and steganography are used as integrated part along with newly developed enhanced security module.

In 2010, Weiqi Luo, Fangjun Huang, and Jiwu Huang presented Edge Adaptive Image Steganography for hiding data in image file [5]. In this work they expanded the LSB matching revisited image steganography and propose an edge adaptive scheme which can select the embedding regions according to the size of secret message and the difference between two consecutive pixels in the cover image.

In Existing System [6-8], the Secret files have been embedded into some standard Cover file formats like images, audio, and video files. Also, when embedding the content they have modified the Cover files binary value in LSB position. MSA Encryption algorithm with some standard randomization methods are used for encrypt the Secret files.

The limitations are the process of embedding the Secret file into the cover file will increase the size of the Cover file. The standard randomization functions, used in the existing system were hard to implement. The system is not fully integrated. LSB and LSB+1 method can be easily cracked. It is more vulnerable to attacks.

3. DESIGN OF PROPOSED SYSTEM

Initially need to find the binary values of cover file and secret file. And encrypt the secret file binary values by using the enhanced MSA algorithm with new randomization methods, and to make the system more secure it introduces the PW (password) for encryption. While encryption value need to precede some randomization methods. After encrypted the file, the binary values has need to embed into the cover file's LSB serious positions and is shown in Fig.1.

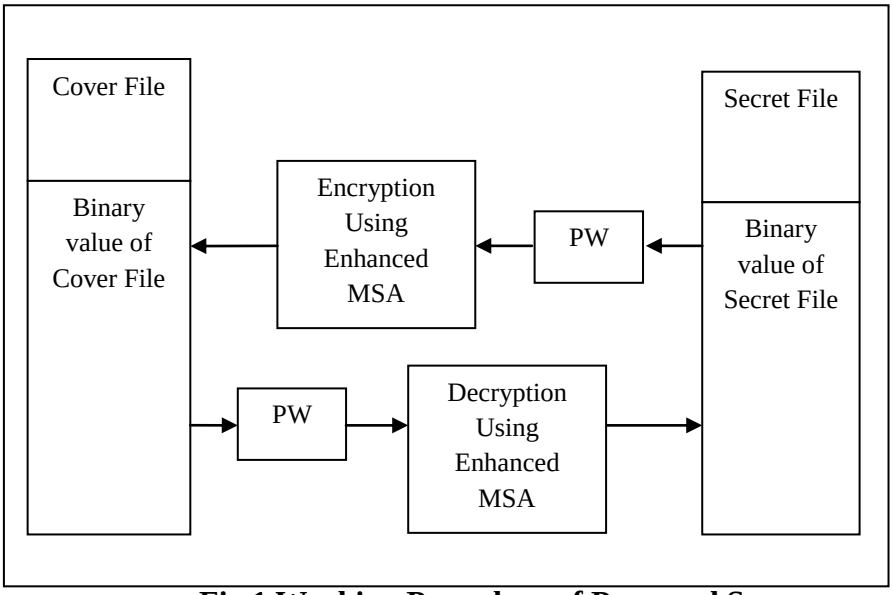


Fig 1 Working Procedure of Proposed System

Embedding the secret file inside the cover file's LSB, LSB+1, LSB+2, LSB+3 positions

To hide the binary values of a secret file inside a cover file, we using cover file's binary value. Each byte of the secret file needed 2 consecutive byte of the cover file, (i.e.) each 8 bit of secret file needed two 8 bit value of cover file. Let us consider two bytes of secret file 10110111 and 11011011, this two byte of information needed 4 bytes values from cover file 0010**1111**, 0011**0111**, 0001**1000**, and 0111**0001**.

When changing partial byte value of the cover file will not affect the original content of the file. And always advised to use the centre positions byte values of the cover files. Because the earlier and the end state of byte values contains the file information. We can use any type non standard files as cover file such as, .exe, .pptx, .docx, .cmd, .xlsx, etc., also the same format of secret file can also be there.

In our proposed system, Randomization methods for more security and time consuming purpose are modified by applying the enhanced MSA algorithm by using different randomization methods over the 16 X 16 Key matrix (i.e. Totally 256 ASCII characters).

3.1 RANDOM KEY GENERATION AND ENHANCED MSA ENCRYPTION ALGORITHM WITH DIFFERENT RANDOMIZATION METHODS

MSA Encryption algorithm is a symmetric key algorithm. It is proposed by Nath et al. [1]. It's using the same key for both encryption and the decryption process. It deals with three parameters for encoding the files, such as randomization number, encryption number and the shift parameter. When decoding the secret file from the cover file user need to give the exact value for the three parameters. Also based on the following table, we need to find the base value for the length of text key.

Table 1 Base value Details for Length of the key

Length of key(L)	1	2	3	4	5	6	7	8
Base value(b)	17	6	15	14	13	12	11	0
Length of key(L)	9	0	11	12	13	14	15	6
Base value(b)	9	8	7	6	5	4	3	2

When encrypting the file, the bytes values of the secret file need to replace with the corresponding characters in the random key matrix. And by using the matrix value need to find the three parameters such as randomization number (P1), encryption number (P2) and shift parameter (P3).
For finding those three, we have to use common sum function

$$\text{Sum} = \sum_{m=1}^{L} (\text{ASCII Code}) * b^{m}$$

By using above equation finding the three parameter for e.g. getting text key with size of 3 from the user = XYZ. Then calculating the sum value for the key "XYZ" is

$$\text{Sum} = (88*15^{1} + 89*15^{2} + 90*15^{3}) = 325{,}095$$

1) *Finding P1: Randomization number*

N1 = 3*1+2*2+5*3+0*4+9*5+5*6 = 97
P1 = Sum mod N1 = (325095) mod (97) = 48
Note: if n1=0 then we set n1=num1 and if n1>64 then n1=n1-64

2) *Finding P2: Encryption number*

N2 = 5*1+9*2+0*3+5*4+2*5+3*6 = 71
P2 = Sum mod N2 = (325095) mod (71) = 57
Note: if n2=0 then we set n2=num2 and if n2>64 then we set n2=n2-64

3) *Finding P3: Relative Shift*

P3 = Sum of all digit in (Sum) = 3+2+5+0+9+5 = 24
By P3 number of times we need to shift the values in the 16 X 16 matrix. By shuffling the values in the matrix will the encoding process stronger.

At previous they have implemented with general 8 steps randomization method. In that method all the 8 randomization method should need to apply in serial manner. It has much more computation process. And it will consume more time.

3.2 RANDOMIZATION METHODS

In this proposing method, we have introduced five different randomization methods. These methods will take minimum level of computation, also less time consuming. Even it having the same level of security level and it easily can able to implement.For making our matrix value to more secure, we need to apply different randomization values over the matrix. As follows:

A. *Function Acyclic()*
B. *Function EvenUpshift()*
C. *Function OddDownshift()*
D. *Function EvenLeftShift()*
E. *Function OddRightShift()*

Every randomization function applying over the matrix value as follows: (The example for the methods has been given in 4X4 text key matrix).

A	B	C	D
E	F	G	H
I	J	K	L

M	N	O	P

Table 2 Original Text Key

A. Function Acyclic(Original Text Key)

In this function, the matrix values as to be shifted in reverse order.

B	C	D	H
A	J	F	L
E	K	G	P
I	M	N	O

Table 3 Acyclic()

B. Function EvenUpshift(Acyclic)

In this function, the **Even** Column of the matrix shifted to one step upper.

B	J	D	L
A	K	F	P
E	M	G	O
I	C	N	H

Table 4 EvenUpShit()

C. Function OddDownShift(EvenUpshift)

In this function, the **Odd** Column of the matrix shifted to one step down.

I	J	N	L
B	K	D	P
A	M	F	O
E	C	G	H

Table 5 OddDownShift()

D. Function EvenLeftShift(OddDownShift)

In this function, the **Even** Row of the matrix shifted to one step left

I	J	N	L
K	D	P	B
A	M	F	O
C	G	H	E

Table 6 EvenLeftShift()

E. Function OddRightShift(EvenLeftShift)

In this function, the **Odd** Row of the matrix shifted to one step right

L	I	J	N
K	D	P	B
O	A	M	F
C	G	H	E

Table 7 OddRightShift()

Repeat the above five steps for P3 number of times for getting more randomization.

In this present work the size of the secret file is much smaller than the cover file. In the cover file, modification to be done in the middle .i.e. Cover file size = 20026 bytes, then modification should start at 10013+1 bytes. And the last few bytes of cover file have been reserved for password, which is given at the time of encryption.

Table 8 Text file bytes embedding into Cover file bytes

Original Text	Bit String	Encrypted Bits to be inserted in LSB positions	Changes in Bit String	Changed Text
X	01011000	0100	01010100	T
Y	01011001	0001	01010001	Q
Z	01011010	0101	01010101	U

In the above example explains that how the text files has been changed after the modification.

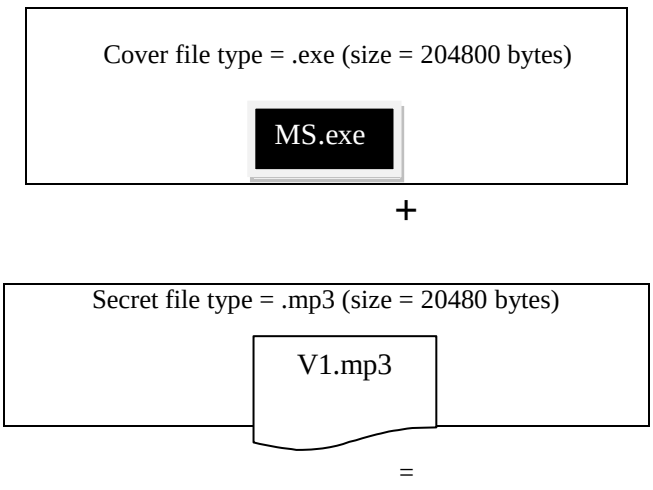
Table 9 Inserting ASCII value byes into cover file

Original Text	Bit String	Encrypted Bits to be inserted in LSB positions	Changes in Bit String	Changed Text
Ö	11110110	0100	11110100	ô
Å	11100101	0001	11100001	á
~	01111110	0101	01110101	u

In the above example explains, that how the ASCII value as been changed after the modification. The encrypted information in cover file by using different randomization methods are hidden. To decrypt the original file from the cover file, reverse the process of randomization with serial manner. The Enhanced MSA algorithm with enhanced algorithm is more secure when compare to the existing system algorithm. It can hide more than two bits. It can use any kind of file format as a cover file. Size of the cover file will remain same when after embedded the secret file.

4. RESULTS AND DISCUSSION

Case 1:



Embedded Cover file = .exe (Size = 204800 bytes)

MS.exe

Case 2:

Cover file type = .docx (size = 36864 bytes)

Steganography

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. The word *Steganography* is of Greek origin and means "concealed writing" from the Greek words *steganos* (στεγανός) meaning "covered or protected", and *graphei* (γραφή) meaning "writing". The first recorded use of the term was in 1499 by Johannes Trithemius in his *Steganographia*, a treatise on cryptography and Steganography disguised as a book on magic. Generally, messages will appear to be something else: images, articles, shopping lists, or some other *covertext* and, classically, the hidden message may be in invisible ink between the visible lines of a private letter.

+

Secret file type = .jpg (Size = 4096 bytes)

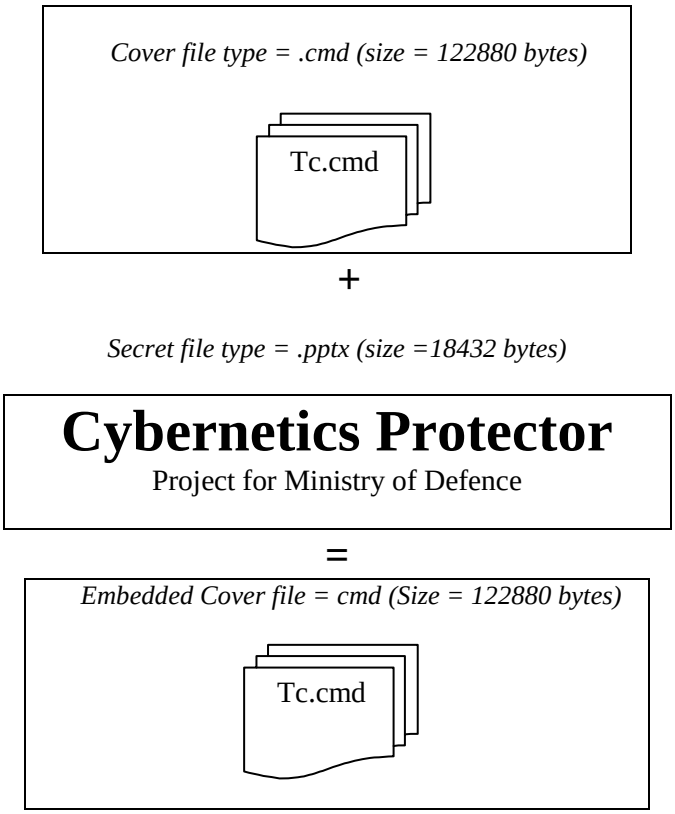
=

Embedded Cover file = .docx (size = 36864 bytes)

Steganography

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. The word *Steganography* is of Greek origin and means "concealed writing" from the Greek words *steganos* (στεγανός) meaning "covered or protected", and *graphei* (γραφή) meaning "writing". The first recorded use of the term was in 1499 by Johannes Trithemius in his *Steganographia*, a treatise on cryptography and Steganography disguised as a book on magic. Generally, messages will appear to be something else: images, articles, shopping lists, or some other *covertext* and, classically, the hidden message may be in invisible ink between the visible lines of a private letter.

Case 3:



5. CONCLUSION

This system is aimed to provide the secure communication and exchanging evidences between the client and server. It is highly recommended for Advanced Steganographics approach. This study has been done over the cryptography and steganography. The data hiding process has been achieved by enhanced MSA algorithm along with different randomization methods. In this technique, security of the secret file would be highly motivated. Because of the variation of randomization function, the present work is more secure than the earlier work. Also, this approach can able to implement into any application level projects. With different and advanced cryptographic algorithm for encryption, finding the user location by using GPS technique, finger impression login, usage of ASCII file format as cover files and decrypt the file by using voice recognition system can be attempted.

6. REFERENCES

[1] Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, "*A Challenge in Hiding Encrypted Message in LSB and LSB+1 Bit positions in various Cover Files"*, Journal of Global Research in Computer Science, ISSN-2229-371X, Vol. 2, No.4 , 2011.

[2] Joyshree Nath and Asoke Nath, "*Advanced Steganography Algorithm using Encrypted secret message*", *(IJACSA)* International Journal of Advanced Computer Science and Applications, Vol. 2, No. 3, 2011.

[3] Dripto Chatterjee, Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, "*Symmetric Key Cryptography using modified DJSSA Symmetric Key Algorithm*", in Proc. of The 2011 World Congress in Computer Science, Computer Engineering and Applied Computing, USA, 2011.

[4] Dipti Kapoor Sarmah and Neha Bajpai, *"Proposed System for Data Hiding Using Cryptography and Steganography",* International Journal of Computer Applications, Vol. 8, No.9, 2010**.**

[5] Weiqi Luo, Fangjun Huang, and Jiwu Huang "*Edge Adaptive Image Steganography Based on LSB Matching Revisited",* IEEE transactions on Information Forensics and Security, Vol.5, No. 2, 2010.

[6] Dripto Chatterjee, Joyshree Nath, Soumitra Mondal, Suvadeep Dasgupta and AsokeNath, "*Advanced Symmetric key Cryptography using extended MSA method: DJSSA symmetric key algorithm", Journal Computing,* Vol. 3, No.2, Page 66-71, Feb 2011.

[7] A.Nath, S.Ghosh and A.Mallik, *"Symmetric key Cryptography using key Generator*", Proceedings of International conference on SAM-2010, Las Vegas(USA), 12-15, Vol.2, pp.239-244, July 2010.

[8] A.Nath, S.Das and A.Chakrabarti, "*Data Hiding and Retrieval",* Proceedings of IEEE International Conference on Computer Network , Bhopal,pp.26-28, Nov 2010.