A New Embedding Algorithm for Data Security

May Htet[1], Su Wai Phyo[2]

[1]Department of Information Technology, Mandalay Technological University, Myanmar

[2]Department of Information Technology, Mandalay Technological University, Myanmar

Abstract: *Nowadays, due to the incredible advance in information and communication technology, security of information transferring through communication channel becomes a major concern. Cryptography and steganography are essential tools for information security. Therefore, in order to achieve more robust security system, this paper focuses on combination usage of cryptographic system and text steganographic technique. Traditionally, all of the steganographic techniques have limited information-hiding capacity. For this reason, this paper proposes a new embedding algorithm to hide a larger amount of secret text in cover text without degrading in the content of the cover. In addition, to enhance the security of secret message, it has been scrambled first through RC5 encryption algorithm before concealing into an innocuous cover text. The basic idea behind this paper is to provide a good, new efficient embedding method for hiding the data in text from hackers and sent to the receiver in a safer manner.*

Keywords: *RC5 Encryption Algorithm, Cryptography, Steganography, Text Steganography, New Embedding Algorithm*

1. Introduction

In today's data communication environments, information security plays a vital role and one of its major concerns is to securely exchange information between the communicators. For this purpose, many information security techniques like cryptography and steganography are used. Cryptography conceals the content of a message, whereas the steganography conceals the existence of a message. Even though both methods provide security, studies have been made to combine both cryptography and steganography methods into one system for better confidentiality and security. The integration of steganography with cryptography provides an extra layer of security that ensures the safe and secure delivery of message to the intended recipient.

Cryptography can be categorized into two: symmetric and asymmetric or public key cryptography. Symmetric algorithms can easily encrypt a vast amount of data than public key algorithms. In this effort, a symmetric cryptographic algorithm, RC5 (Rivest Cipher 5) is used for enciphering the secret data.

Steganography can be classified into text, image, audio, and video steganography depending on the cover media used to embed secret data. In this paper, only steganography in text file is considered and a new efficient embedding method is employed for hiding the encrypted message into the text file to obtain more efficient information security system.

2. Related Works

Many research areas proposed hybrid security systems in order to overcome the security problems. Most of these systems focused on the combination of cryptographic and steganographic techniques according to their security requirements.

In the previous research [1], the authors proposed an efficient method of text steganography by using Word Mapping Method and genetic operator crossover for generating the encrypted form of the message. The main aim of this paper is to hide encrypted data in text file in order to achieve high level of security. Then, Indradip Banerjee and others presented an information hiding system using a text quantum steganography technique. In their paper, new code representation technique (SSCE) has been used to produce the encrypted message. This approach is capable of secure transfer of the message to the authorized recipient [2]. Moreover, Niimi and his fellows suggested a data hiding technique so called Semantic Method by using synonyms of certain words to

http://dx.doi.org/10.17758/ERPUB.R815204

obscure the actual data content, thereby hiding the information in the text for achieving higher security and rubustness [3]. In [4], Monika Agarwal presents three novel approaches of text steganography. To obtain a more secure system, one-time pad scheme is used before data hiding. Thus, the proposed models give two layers of security. In addition, the author concluded that the proposed approaches achieve better embedding capacity than some other existing approaches.

According to literature and concepts from previous works, this work is intended to provide a new powerful embedding technique for secret data hiding in text based on Hiding Data in Paragraph (HDPara) algorithm in order to get better embedding capacity. In addition, with the aim of satisfying the confidentiality requirement of information security, symmetric cryptographic algorithm is combined with the new proposed embedding technique.

3. Text Steganography

Text steganography is a steganography technique that uses text as the cover media for burying the secret information. Among different types of steganography, text steganography is most tricky due to the lack of redundant information in text files to hide a secret message as compared to other media. However, storing text file require less memory and its faster as well as easier communication makes it preferable to other types of steganographic methods. The process of embedding information inside a text file can be represented in simple equation as [5]:

$$\text{Cover Text + Secret Information (text/image/audio format) = Stego Text} \quad (1)$$

Text steganography can be broadly classified into three types: format-based, random and statistical generation, and linguistic methods. There have been tremendous researches in the field of text steganography based on these three categories. In addition, many researchers have investigated other types of text steganographic techniques for hiding information in text. Hiding Data in Paragraph (HDPara) algorithm and the new proposed embedding algorithm (OHDPara) are based on other popular text steganographic techniques.

3.1 Hiding Data in Paragraph (HDPara) Algorithm

This approach makes use of a pre-determined cover file which can be any meaningful piece of English text and can be drawn from any source (e.g., a paragraph from a newspaper/book). The approach works by hiding a message using start and end letter of the words of a cover file. It works on the binary value of a character as opposed to the other approaches which work on the ASCII value [4].

After converting the cipher text to a stream of bits, each bit is hidden by picking a word from the cover file and using either the start or the end letter of that word depending on the bit to be concealed. Bit 0 or 1 is hidden by reading a word, sequentially, from the cover file and writing it down in the stego file and positioning the start or the end letter, respectively, of the word in the stego key. A word having same start and end letter is skipped in hiding process.

To hide bit 0, write the start letter of the word in the stego key. If the secret bit is 1, write the end letter of that word in the stego key. The process is repeated till the end of binary file. Since no change is made to the cover, the cover file and its corresponding stego file are exactly the same.

As an example, it is considered a message “Try hard!” to be concealed in a cover file. The cipher text of the message was found to be “W*μ*ĉı̣&d*μ*tP”. Fig. 1, 2, 3, and 4 show the cover file, the stego file, the stego key, and the embedding structure of HDPara algorithm, respectively. Before concealing, convert the message into binary stream.

Lights for navigation have existed for more than three thousand years. Their purpose has been to show ships where they are and to guide them into safe harbours or to warn them of rocks and reefs that could destroy them. Although preventing loss of life has always been a consideration, it is the preservation of ships and cargoes that has been the real driving force behind lighthouse construction.

Fig. 1: Cover File

Lights for navigation have existed for more than three thousand years. Their purpose has been to show ships where they are and to guide them into safe harbours or to warn them of rocks and reefs that could destroy them. Although preventing loss of life has always been a consideration, it is the preservation of ships and cargoes that has been the real driving force behind lighthouse construction.

Fig. 2: Stego File

sfeermtedsTpsbowsetadogtishrrwmorastcdtAgsfe
hanctstpfsasthbtl

Fig. 3: Stego Key

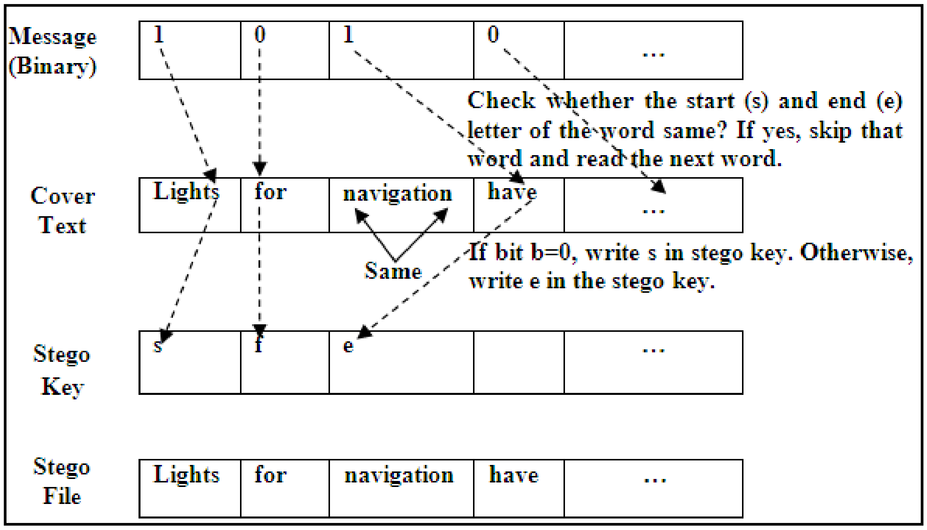

Message (Binary)
1
0
1
0
...
Check whether the start (s) and end (e) letter of the word same? If yes, skip that word and read the next word.
Cover Text
Lights
for
navigation
have
...
Same
If bit b=0, write s in stego key. Otherwise, write e in the stego key.
Stego Key
s
f
e
...
Stego File
Lights
for
navigation
have
...

Fig. 4: Embedding Structure of HDPara Algorithm

3.2 New Embedding Algorithm (OHDPara)

The new embedding algorithm is based on Hiding Data in Paragraph (HDPara) algorithm. This approach also uses pre-defined cover text file and works on the binary value of a character. It performs message hiding by picking a word sequentially from the cover and using either first, second, second last, or last letters of the words depending on the bit to be secreted.

Firstly, read a bit from the binary (bin) file. Then read a word from the cover text and write it in the stego file. Subsequently, check whether the first and last letter of the word is same. A word having same first and last letter is skipped in hiding process. If the hidden bit is 0, write the first letter of the word in the stego key. Or else, write the last letter of the word in the stego key.

Next, check whether the second and second last letter of the word is same. If they are same, skip that word and pick another word for hiding successive bit from bin file. If not, the following bit is hidden using either the second or second last letter of the same word. Write the second letter in the key file if the secret bit is 0 and otherwise, write the second last letter in the stego key. The hiding process is repeated till the end of the bin file. If the size of cover file is not enough for hiding the secret data, read the word from the beginning of the cover file. However, for the next times, the words are not written again in the stego file to generate uniformly the same stego file as the cover file.

As an example, it is considered a secret message “Try hard yourself!” to be concealed in the same cover file. After enciphering the message, the cipher text generated was “Wµĉ¦&dµtʾïĉvψ¶ĢxгP”.

Lights for navigation have existed for more than three thousand years. Their purpose has been to show ships where they are and to guide them into safe harbours or to warn them of rocks and reefs that could destroy them. Although preventing loss of life has always been a consideration, it is the preservation of ships and cargoes that has been the real driving force behind lighthouse construction.

Fig. 5: Cover File

Lights for navigation have existed for more than three thousand years. Their purpose has been to show ships where they are and to guide them into safe harbours or to warn them of rocks and reefs that could destroy them. Although preventing loss of life has always been a consideration, it is the preservation of ships and cargoes that has been the real driving force behind lighthouse construction.

Fig. 6: Stego File

sirhvexrerthehdnsrTheshbtsherteaatedthinefsrotnameorkar
fcodetehggrlsfeissynntstnrfdsehberagnfodnescLtfevdxreot
aehtnserhpuhn

Fig. 7: Stego Key

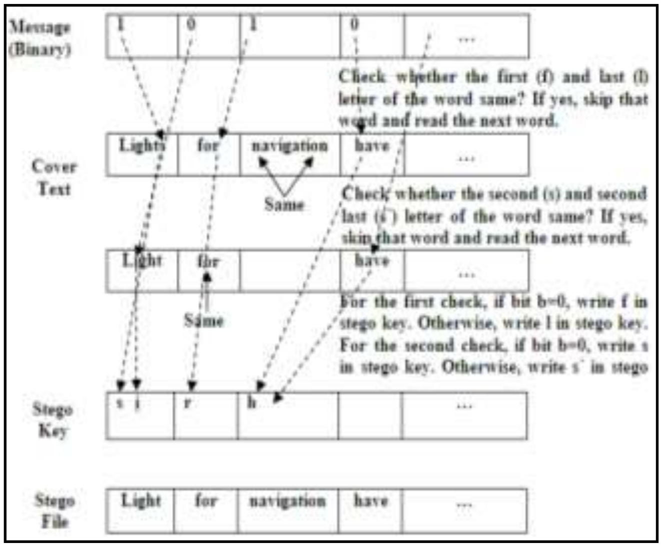

Message (Binary)
1
0
1
0
...
Check whether the first (f) and last (l) letter of the word same? If yes, skip that word and read the next word.
Cover Text
Lights
for
navigation
have
...
Same
Check whether the second (s) and second last (s´) letter of the word same? If yes, skip that word and read the next word.
Light
for
have
...
Same
For the first check, if bit b=0, write f in stego key. Otherwise, write l in stego key. For the second check, if bit b=0, write s in stego key. Otherwise, write s´ in stego
Stego Key
s
i
r
h
...
Stego File
Light
for
navigation
have
...

Fig. 8: Embedding Structure of OHDPara Algorithm

http://dx.doi.org/10.17758/ERPUB.R815204

4. RC5 Algorithm

RC5 is a fast and simple symmetric block cipher that uses the same key for encryption and decryption. The plaintext and ciphertext are fixed-length bit sequences. A novel feature of RC5 is the heavy use of data-dependent rotations. RC5 has a variable word size “w” (16, 32, and 64); a variable number of rounds “r” (0-255) and a variable length secret key “b” (0-255). The notation of the RC5 parameters is RC5-w/r/b.

RC5 algorithm is divided into three parts:

- Key Expansion Algorithm
- Encryption Algorithm
- Decryption Algorithm

RC5 uses an expanded key table S[0...t-1], consisting of t = 2(r+1) words, where, r is the number of rounds. The key expansion algorithm initializes S from the users given secret key parameter K by using two magic constants to perform initialize operation and consists of three simple algorithmic parts. The two magic constants are two word-sized binary constents, P_w and Q_w.

TABLE I: Magic Constant Values in Hexadecimal Form

w	16	32	64
P_w	b7e1	b7e15163	b7e15128aed2a6b
Q_w	9e37	9e3779b9	9e3779b97f4a7c15

Three simple algorithmic parts are shown in Fig. 9. Fig. 10 illustrates the operation of RC5 algorithm.

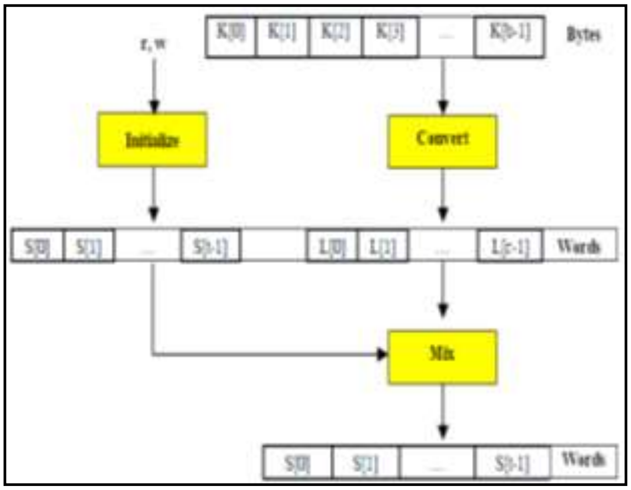


Fig. 9: Key Expansion Process of RC5

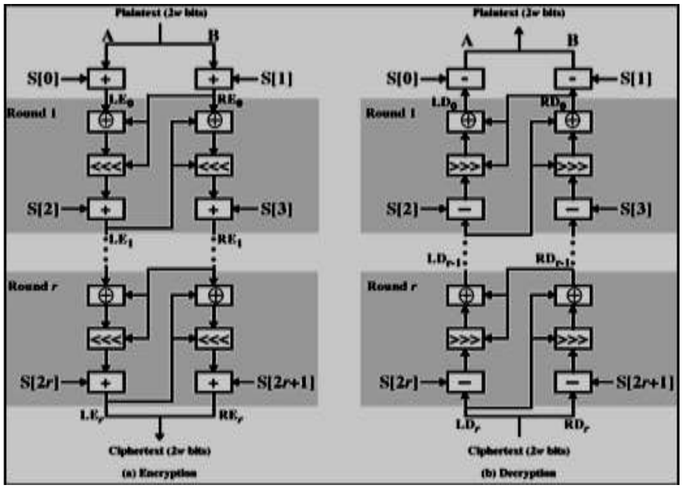


Fig. 10: Operation of RC5 Algorithm

TABLE II: RC5 Encryption and Decryption Algorithms

Encryption Algorithm	Decryption Algorithm
Input: Plaintext stored in two w-bit input registers A and B, Number r of rounds, Key array S[0,…, t-1]	Input: Ciphertext stored in two w- bit input registers LE_r and RE_r, Number r of rounds, Key array S[0,…, t-1]

Procedure:
LE_0 = A + S[0];
RE_0 = B + S[1];
for i = 1 to r do
$LE_i = (LE_{i-1} \oplus RE_{i-1}) <<< RE_{i-1}) + S[2 * i]$;
$RE_i = ((RE_{i-1} \oplus LE_i) <<< LE_i) + S[2 * i + 1]$;
	Procedure:
	for i = r downto 1 do
	$RD_{i-1} = ((RD_i - S[2 * i + 1]) >>> LD_i) \oplus LD_i$;
	$LD_{i-1} = ((LD_i - S[2 * i] >>> RD_{i-1}) \oplus RD_{i-1}$;
	B = RD_0 - S[1];
	A = LD_0 - S[0];

Output: Ciphertext stored in registers LE_r and RE_r	Output: Plaintext stored in registers A and B

5. Proposed System Architecture

The proposed system architecture is organized with two portions: sender side which consists of encryption section and embedding section and receiver side which consists of decryption section and extraction section as illustrated in Fig. 11 and Fig. 12.

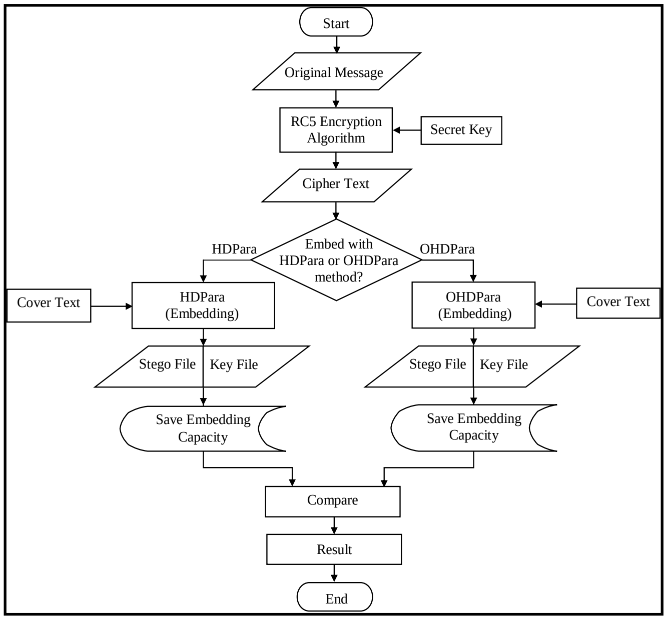


Fig. 11: Processes of the Sender Side

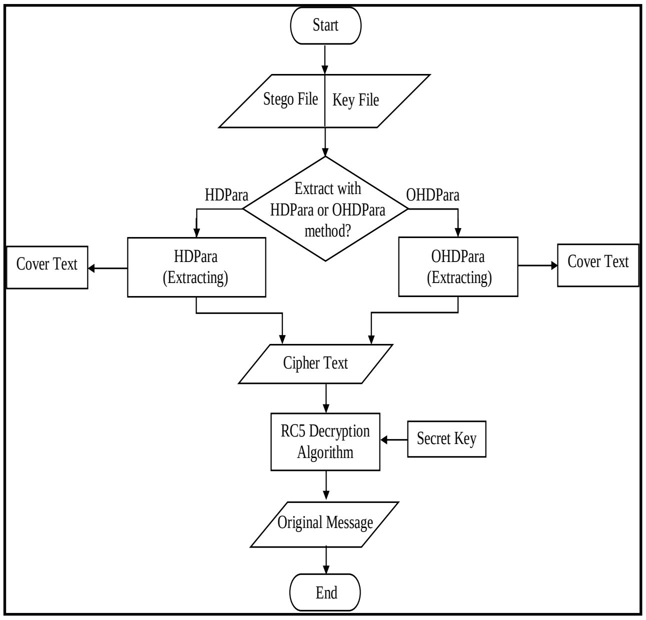


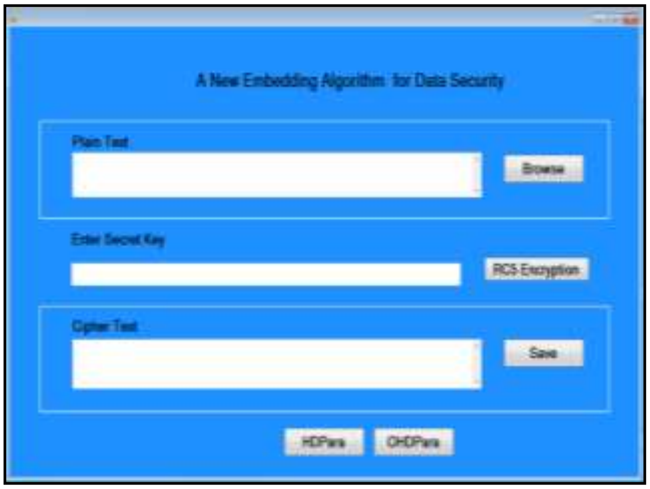
Fig. 12: Processes of the Receiver Side

In Fig. 11, original message or plaintext is firstly loaded and encrypted using RC5 encryption algorithm. Then, the generated cipher text is embedded into the cover text using both HDPara and OHDPara embedding methods. After that, the embedding capacity of the two algorithms are saved and compared to show the final result which algorithm provides better data hiding capacity.

As illustrated in Fig. 12, the receiver performs the extraction of cipher text from stego text with the help of stego key by using either HDPara or OHDpara embedding method. Then, the original message is recovered by decrypting the cipher text with RC5 decryption algorithm.

6. System Implementation

The proposed system implementation is represented as a series of interface. The main interface of the proposed scheme is illustrated in Fig. 13.

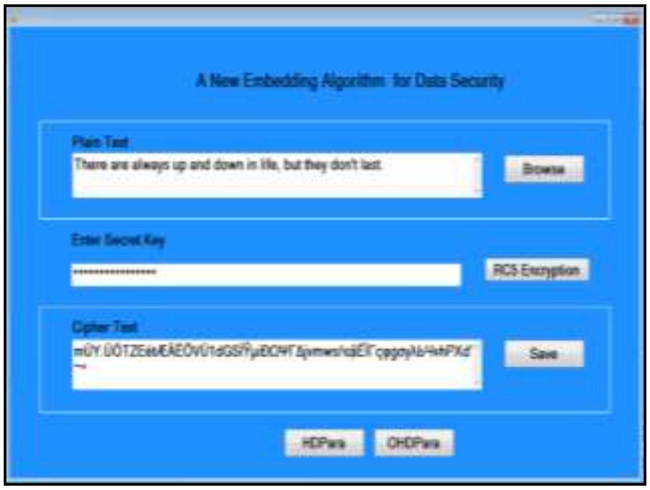

A New Embedding Algorithm for Data Security

Plain Text

Enter Secret Key

Cipher Text

Fig. 13: Main Interface


A New Embedding Algorithm for Data Security

Plain Text: There are always up and down in life, but they don't last.

Enter Secret Key: ******************

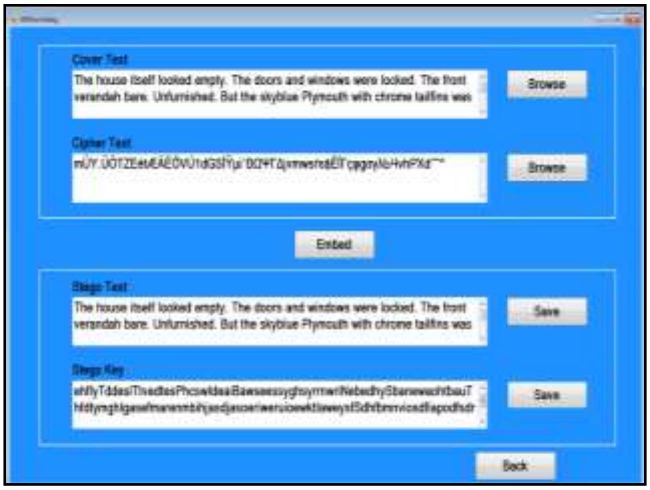
Cipher Text: mÜY.ÙÖTZEèéÆÉÒVÜ1dGSŸµÐO¥TΔjvmws\ciÈΓcpgqyÀb¼#PXd' ¬

Fig.14: Encryption Page

The user can load the original message and encrypt it with the help of secret key and RC5 encryption algorithm as shown in Fig. 14. After that, the user can select one of the two embedding algorithms to embed the cipher text by applying "HDPara" or "OHDPara" button.

If the user clicks the "HDPara" button, embedding page using HDPara algorithm will be appeared and the cipher text is embedded into the cover text using HDPara algorithm to generate stego text and stego key as shown in Fig. 15.

If the size of the cover file is not enough for hiding the whole cipher text, Fig. 16 will be appeared to inform the user to choose another cover file.



Cover Text

The house itself looked empty. The doors and windows were locked. The front verandah bare. Unfurnished. But the skyblue Plymouth with chrome tailfins was

Cipher Text

[illegible]

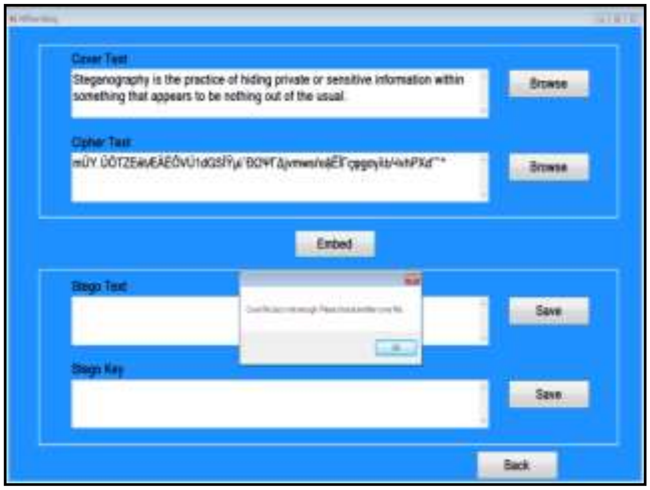
Stego Text

The house itself looked empty. The doors and windows were locked. The front verandah bare. Unfurnished. But the skyblue Plymouth with chrome tailfins was

Stego Key

[illegible]

Fig. 15: Embedding Using HDPara Algorithm



Cover Text

Steganography is the practice of hiding private or sensitive information within something that appears to be nothing out of the usual.

Cipher Text

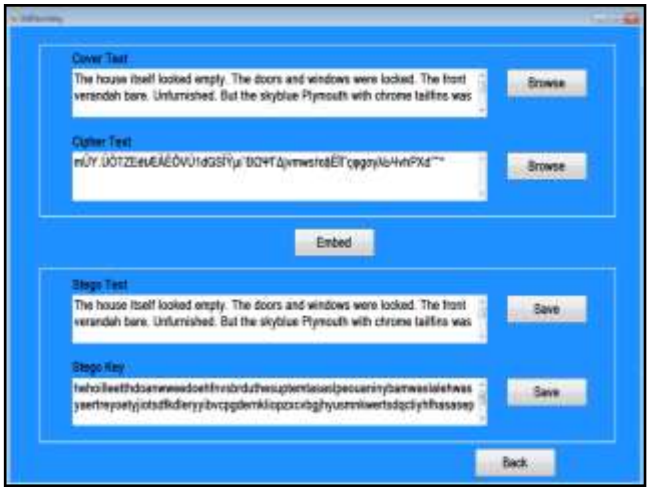
[illegible]

Stego Text

Stego Key

Fig. 16: Information for Invalid Cover File

Like the embedding process of HDPara, OHDPara embedding is performed as shown in Fig. 17.



Cover Text

The house itself looked empty. The doors and windows were locked. The front verandah bare. Unfurnished. But the skyblue Plymouth with chrome tailfins was

Cipher Text

[illegible]

Stego Text

The house itself looked empty. The doors and windows were locked. The front verandah bare. Unfurnished. But the skyblue Plymouth with chrome tailfins was

Stego Key

[illegible]

Fig. 17: Embedding Using OHDPara Algorithm

At the receiver side, extracting process and decryption process can be performed by using the reverse orders of the sender's processes to obtain the original message.

7. Experimental Results

According to the experiment, it can be considered at the two parameters: security consideration and consideration on embedding capacity.

7.1 Security Consideration

In the information hiding process of the proposed scheme, a new embedding technique is used in order to provide more reliability and robust security. In this proposed approach, four specific letters are used instead of two specific letters for hiding the secret bit. Thus, at the security point of view, it is more difficult for anyone to guess how to extract the secret message from the stego text without knowing the stego key. In addition, a symmetric block cipher, RC5 is used to encrypt the secret data before hiding into the cover text file. Therefore, even if the existence of covert communication can be determined by attackers, the encoding or the obfuscation of data complicates the retrieval of data, typically requiring access to keys.

7.2 Consideration on Embedding Capacity

Capacity is defined as the ability of a cover media to hide secret information. The capacity ratio is computed by dividing the amount of hidden bytes over the size of the cover text in bytes [6].

$$\text{Capacity ratio} = \frac{\text{amount of hidden bytes}}{\text{size of the cover text in bytes}} \tag{2}$$

Assuming one character occupies one byte in memory, the percentage capacity which is capacity ratio multiplied by 100 have calculated.

In order to prove that the proposed approach is better than the HDPara approach in terms of data embedding capacity, tests are carried out based on ten experimental sample data.

Examples of secret message used are

1. “Try” (3 bytes)
2. “Second” (6 bytes)
3. “Hello World!” (12 bytes)
4. “Success comes after work!” (25 bytes)
5. “There are always up and down in life, but they don't last.” (58 bytes)
6. “Networking as a concept has acquired what is in all truth an unjustified air of modernity.” (90 bytes)
7. Cryptography scrambles a message to conceal its content, whereas stegnography conceals the existence of message.” (112 bytes)
8. “There is a legend that St Augustine in the fourth century AD was the first individual to be seen reading silently rather than a loud, or semi-aloud, as had been the practice hitherto. Reading has come a long way since Augustine's day.” (233 bytes)
9. “Steganography is the practice of hiding private or sensitive information within something that appears to be nothing out of the usual. Steganography is often confused with cryptology because the two are similar in the way that they both are used to protect important information. The difference between the two is that Steganography involves hiding information.” (361 bytes)
10. “Steganography is the art and science of data hiding. In contrast with cryptography, which secures data by transforming it into another, unreadable format, steganography makes data invisible by hiding (or embedding) them in another piece of data, known alternatively as the cover, the host, or the carrier. The modified cover, including the hidden data, is referred to as a stego object. It can be stored or transmitted as a message.” (432 bytes)

Table 3 compares the percentage capacity of HDPara and OHDPara approaches over ten example data. In HDPara algorithm, it is needed to consider the size of cover text which is enough for hiding the data. In contrast, in proposed algorithm, even if larger secret message is hidden, there is no need to consider the larger cover size as it reads the word from the beginning of the cover text repetitively. Therefore, it can be clearly seen that the larger the size of secret message, the greater the percentage capacity of the proposed approach.

TABLE III: Percentage Capacity of the Two Approaches

Secret Message	1	2	3	4	5	6	7	8	9	10
HDPara	1.38	2.11	2.34	2.27	2. 35	2.39	2.36	2.39	2.39	2.40
OHDPara	3.33	2.75	3.36	5.94	13.78	21.38	26.60	55.34	85.75	102.61

Table 4 compares the average percentage capacity of the new proposed approach with the HDPara approach (having % capacity > one).

TABLE IV: Average Percentage Capacity of the Approaches

Algorithm	Average % Capacity
HDPara	2.24
OHDPara	32.08

8. Conclusion

A new embedding algorithm based on HDPara algorithm is presented in this paper. In OHDPara algorithm, some of the additional embedding steps are added to the original algorithm in order to get more efficient embedding capacity.

Like HDPara algorithm, the optimized algorithm does not make use of extra white spaces or misspelled words to hide secret data; it works by using specific character of words of any natural looking meaningful piece of English Text. Embedding is done in such a way that there is no or minimum degradation in the content of the cover. Thus, stego files will not draw suspicion regarding the existence of hidden information and it cannot be easily detected by third party. According to the results shown in Table 3 and 4, it can be obviously recognized that the proposed OHDPara algorithm is more efficient than original HDPara algorithm in terms of data embedding capacity. This system is suited for digital data transmission through internet and other information exchange systems. It can be applied in various data communication environments for data security.

As further extensions, the concept of proposed approach can also be applied in other languages. In addition, new embedding techniques can be developed for more efficient data hiding based on the other embedding algorithms and other media cover format.

9. Acknowledgement

The author would like to thank to Dr. Myint Thein, Rector of Mandalay Technological University, for his motivation, guidance, and support. The author is particularly grateful to Dr. Aung Myint Aye, Associate Professor and Head of Department of Information Technology, Mandalay Technological University, for his support, and guidance. The author would like to express her heartfelt gratitude to her supervisor Dr. Su Wai Phyo, Associate Professor, Department of Information Technology, Mandalay Technological University, for her kind advice, permission, and supervision. Finally, the author would like to thank to Dr. May Zin Oo, Lecturer, Department of Information Technology, Mandalay Technological University, for her kind support and encouragement in writing and submitting of this paper.

10. References

[1] Souvik Bhattacharyya et al. A Novel Approach of Secure Text Based Steganography Model using Word Mapping Method (WMM). International Journal of Computer and Information Engineering 4:2 2010.

[2] Banerjee, S. Bhattacharyya, and G. Sanyal, Novel text steganography through special code generation, Int. Conf. on Systemics, Cybernetics and Informatics, 2011, pp. 298-303.

[3] M. Niimi, S. Minewaki, H. Noda, and E. Kawaguchi, A Framework of Text-based Steganography Using SD Form Semantics Model, Pacific Rim Workshop on Digital Steganography 2003, Kyushu Institute of Technology, Kitakyushu, Japan, July 3-4, 2003.

[4] Monika Agarwal. Text Steganographic Approaches: A Comparison. International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1, January 2013.

[5] Mohammad Pooyan, Ahmed Delforouzi, LSB-based Audio Steganography Method Based on Lifting Waveletransform,International Symposium on Signal Processing sand Information Technology, IEEE, 2007.

[6] F. A. Haidari, A. Gutub, K. A. Kahsah, and J. Hamodi, Improving security and capacity for Arabic text steganography using kashida" extensions, IEEE/ACS Int. Conf. on Computer Systems and Applications, 2009, pp. 396-399.