Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

An Approach to Hiding of Encrypted Text Information behind Word Document File

[1]*Neha Gupta*, [2]**Prof. Manoj Sharma**

RGPV Bhopal, India

Abstract: ***The need to secure computer systems is well understood and securing data must be part of an overall computer security plan. Growing amounts of sensitive information are being retained in files and more of these files are being made accessible via the Internet. As more information is made available electronically/digitally, it can be assumed that vulnerabilities and threats to the integrity of that information will increase as well. Information security is becoming an increasingly important area and need to develop core understandings in this area. The primary objectives of information security are to prevent unauthorized access to information, prevent unauthorized tampering or modification of information, and to insure that information remains available when needed. The concepts related to information security are multifaceted. People need to use the cryptographic operations in order to keep the personal sensitive information files to avert from foreigners in consideration of the security. The algorithm include bunch of to protect the attacks of foreigners to reach and read personal files that is located in personal computer or the owner would like to send somewhere or other location. Cryptographic consist of encryption and decryption techniques in computer and computer networking. In effort to keep information's in safety such as banking account information's or to provide file transaction without any problem such as password sharing caused such a security methods. This report describes what features are in the scope of the research and what are not in the scope of the research to be developed. Basically proposed work is dual security approach over confidential information. In this text information encrypted through encryption process then this encrypted information hide behind a word file. At other end extract encrypted information from encrypted cover file and then decrypted these information through decryption process to get original information.***

Keywords: Information security, Encryption, Decryption, Cryptography, Symmetric, Asymmetric, Key, Algorithm

Section I - Introduction

The importance of security in this paper has greatly increased over the years as most of critical functionality of the business and military enterprises became digitized. Text information is an integral part of any information system and they often hold sensitive data. The security of the information depends on physical security, OS security and DBMS security. Information security can be compromised by obtaining sensitive information, changing information or degrading availability of the information. Though access control model were developed and found to ensure security, there were always chances of those access controls to be bypassed leading to a breach. To enforce the second layer of security, data being stored in the repository could be modified and stored in an encrypted format. This idea gave way to research in the design possibilities of information. Two of such designs were Access Control Kernels and Encrypted information. Access Kernels were based on isolating and containing security policies inside separate modules. The downside of this design was that the value-dependent access restrictions were not possible. The cryptographic technique of using keys to encrypt and store information was applied to achieve security. There were many restrictions and challenges like operations\computations on encrypted data, view-based protection, etc. This paper focused on a security solution for protection of data/information, specifically protection of data/information that resides in files. Mostly information are deployed and stored in some kind of a persistent storage device like a disk drive. Periodically, even in-primary memory (RAM) database have the need to backup data/information; hence data could end up in a persistent storage device in plaintext. Encryption, the process of disguising data/information in such a way to hide its content which is a highly effective way to achieve security for data at rest the need to secure computer systems is well understood and securing data must be part of an overall computer security plan. Growing amounts of sensitive information are being retained in files and more of these files are being made accessible via the Internet. Information security is becoming an increasingly important topic and need to develop core understandings in this area. The primary objectives of information security are to prevent unauthorized access to information, prevent unauthorized tampering or modification of information/data, and to insure that information/data remains available when needed. The concepts related to information security are multifaceted. Basically proposed work is dual security approach over confidential information. In this text information encrypted through encryption process then this encrypted information hide behind a word file. At other end extract encrypted information from encrypted cover file and then decrypted these information through decryption process to get original information. This paper is dividing in four sections. Section-I, presents introduction about the work on

security, Section-II, presents proposed work; Section-III presents results analysis of the proposed technique and Section-IV, presents conclusion and references.

Section II - Proposed Work

2.1 Proposed Concept: As we know that in cryptography there are two ways to encrypt information one is symmetric key and another is asymmetric key concept. As we know that symmetric key concept is suitable to encrypt large information as compare asymmetric key concept because symmetric key 1000 time faster as compare asymmetric key concept. So proposed encryption concept is based on symmetric key concept. Figure 2.1 is showing the block diagram of proposed concept. The proposed concept is dual security level concept, one is symmetric block cipher cryptography concept where plain text encrypted/decrypted in the form of block of character at a time and another is encrypted text hiding concept that means encrypted information will hide behind a word file [1]. Due to this security of the information become very high. In this encrypted text file and cover file (word file) will execute with each other. During hiding process selected cover file (word file) should be double in size so the encrypted information can be hide easily because blank space of the selected cover file (word file) will used to hide encrypted information. The presented work proposes a new method for hiding any encrypted secret message inside a text/ASCII or Microsoft word file, through manipulating the white and blank space characters of a cover file. Initially the confidential message is encrypted using proposed encryption process. For hiding confidential message within cover file (word file) I am using technique define in [1]. The main objective of the proposed work is to provide improved security to confidential data while it is being transferred through the users, first they encrypt and after that they hiding it as well.

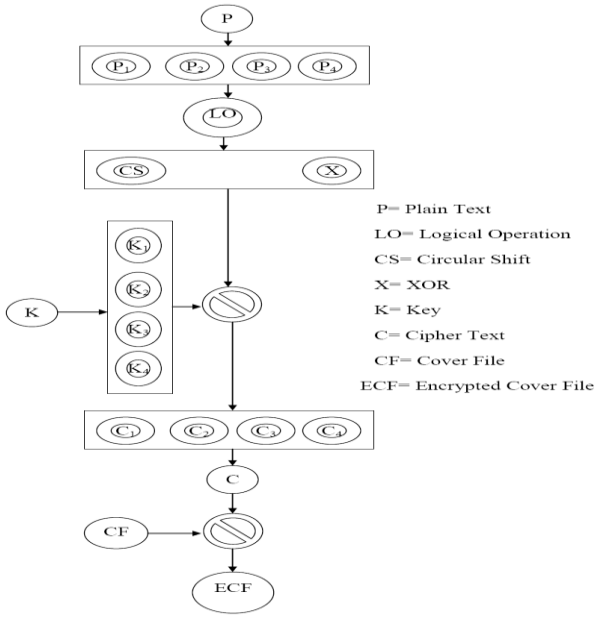


Figure 2.1: Block Diagram of Proposed Concept

2.2 Key Selection Process: Figure 2.2 is showing the architecture of proposed key selection process. According to proposed concept original key is dividing into four sub keys and during encryption process these sub key will use, so the sequence of the sub key is very important. If sequence of the sub keys change then cipher information will not decrypted. To improve security we have proposed a new approach for key selection. Basically this architecture is providing to chosen sub key sequence to form a single private. In this approach a number between 4 to 8 selected randomly and this number dividing from 2. After that this will produced two numeric values one is float and another is integer value. Float value having two choices. Based on these value sequences of the sub keys will decided. Similarly Integer value also having three choices which will give the sequence of the sub keys. The detailed description of proposed technique is as follow:

1. Input a number (N) between 4 to 8 randomly.
2. Calculate the Ratio (R) of Input Number (N) and 2.
   R = N/2
3. Check the Type of Ration (R)
   (R == Float ? Integer)
4. If R is equal Float then Check Value of Ratio (R)
   (R == 3.5) then the sequence of sub Keys is
   K_4, K_1, K_2 and K_3
   Else
   The Sequence of sub keys is
   K_2, K_3, K_4 and K_1
5. If R is equal Integer then Check Value of Ratio (R)

If (R == 2) then the sequence of sub Key is
K_1, K_2, K_3, and K_4
Else if (R == 3) then the sequence of the sub keys is
K_3, K_4, K_1 and K_2,
Else
The Sequence of the sub keys is
K_2, K_1, K_4 and K_3,

6. Exit.

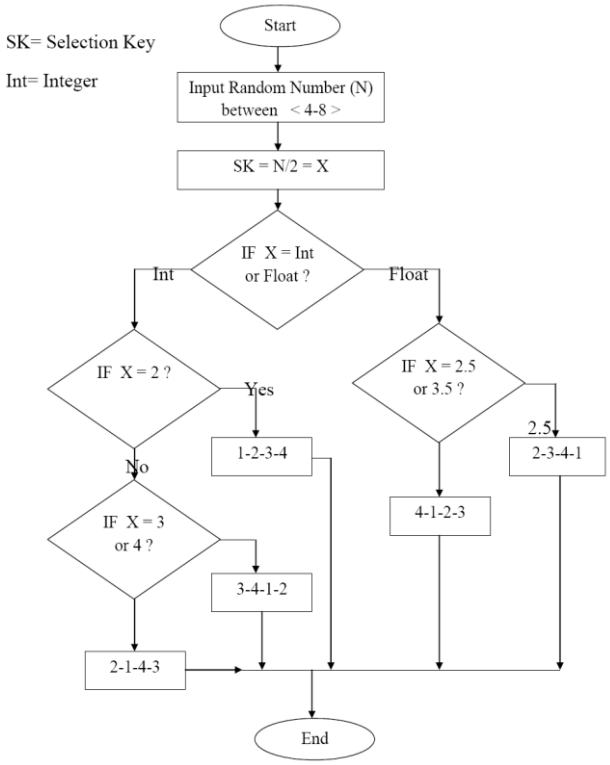


Figure 2.2: Proposed Key Selection Architecture

2.3 Proposed Encryption Architecture: Figure 2.3 is showing proposed encryption architecture. Initially 128 bits or 16 characters plain text selected which is dividing into four sub parts. Each sub parts are performing logical operation like right circular shift and logical operation with sub keys. Detailed descriptions of each operation are shown in the figure 2.3. And step of the encryption algorithm is defined in next section 2.3.1.

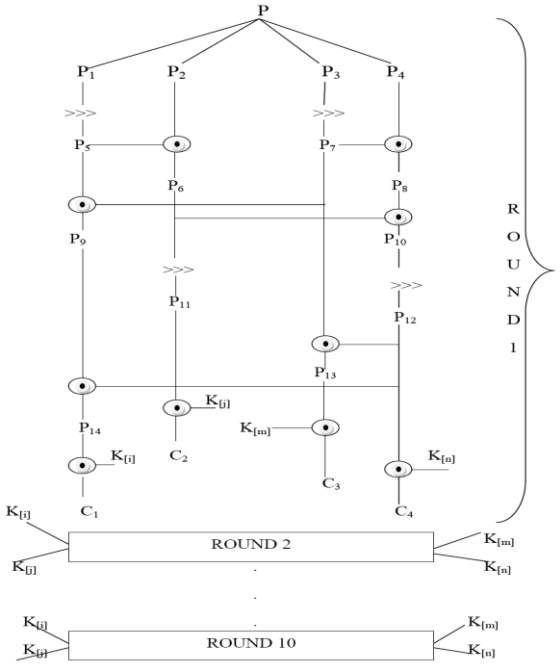


Figure 2.3: Proposed Encryption Architecture

2.3.1 Encryption Algorithm Steps:

1. Input 128 bits plain text (P).
2. Input 128 bits Key (K).
3. Divide Key (K) into four sub Keys like K_1, K_2, K_3 and K_4 each of 32 bits in size
4. Check sequence of sub keys (use Key selection process. See section 4.2)
   K_i, K_j, K_m, and K_n
5. Divide plain text (P) into four sub parts like P_1, P_2, P_3 and P_4 each of 32 bits in size.
6. Apply 3 bits right circular shift on P_1 and P_3. This will produced P_5 and P_7 respectively.
7. Perform XOR operation between P_5 & P_2 and P_7 & P_4. This will produced P_6 and P_8 respectively.
8. Apply XOR operation between P_7 & P_5 and P_6 & P_8. This will produced P_9 and P_{10} respectively
9. Apply 3 bits right circular shift on P_6 and P_{10}. This will produced P_{11} and P_{12} respectively.
10. Perform XOR operation between P_7 & P_{12} and P_{12} & P_9. This will produced P_{13} and P_{14} respectively.
11. Now use sub keys K_i, K_j, K_m and K_n to perform XOR operation with P_{11} P_{12} P_{13} and P_{14} in following way
    K_i XOR P_{14} → C_1
    K_j XOR P_{11} → C_2
    K_m XOR P_{13} → C_3
    K_n XOR P_{12} → C_4
12. Combine C_1, C_2, C_3 and C_4 into cipher text (CT).
13. Repeat process 10 times.
14. Exit.

2.4 Proposed Decryption Architecture: Figure 2.4 is showing architecture of proposed decryption process. Initially it is taking 128 bits or 16 character cipher text which is dividing into four sub parts equally of 32 bits. Then each sub part is performing logical operation like reverse right circular shift and XOR operation with sub keys. Detailed descriptions of each operation are shown in the figure 2.4. And step of the decryption algorithm is defined in next section 2.4.1.

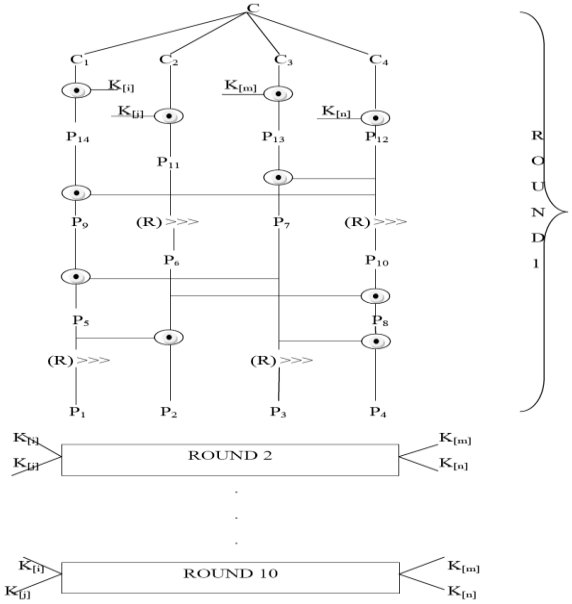


Figure 2.4: Proposed Decryption Architecture441

2.4.1 Decryption Algorithm Steps:

1. Input 128 bits cipher text (CT).
2. Input 128 bits Key (K).
3. Divide Key (K) into four sub Keys like K_1, K_2, K_3 and K_4 each of 32 bits in size
4. Check sequence of sub keys (use Key selection process. See section 4.2)
   K_i, K_j, K_m, and K_n
5. Divide cipher text (CT) into four sub parts like C_1, C_2, C_3 and C_4 each of 32 bits in size.
6. Now use sub keys K_i, K_j, K_m and K_n to perform XOR operation with C_1 , C_2 , C_3 and C_4 in following way
   K_i XOR C_1 → P_{14}
   K_j XOR C_2 → P_{11}
   K_m XOR C_3 → P_{13}
   K_n XOR C_4 → P_{12}
7. Perform XOR operation between P_{12} & P_{13} and P_{12} & P_{14}. This will produced P_7 and P_9 respectively.
8. 8. Apply 3 bits reverse right circular shift on P_{11} and P_{12}. This will produced P_6 and P_{10} respectively.
9. Perform XOR operation between P_7 & P_9 and P_6 and P_{10}. This wills Produce P_5 and P_8 respectively.
10. Apply 3 bits reverse right circular shift on P_5 and P_7. This will produced P_1 and P_3 respectively.
11. Perform XOR operation between P_5 & P_6 and P_7 & P_8. This will produced P_2 and P_4 respectively.

12. Now combine P_1, P_2, P_3 and P_4 into plain text (P).
13. Repeat process 10 times
14. Exit

2.5 Features of the Presented Algorithm: There are many features of the proposed encryption/decryption process which is follows: proposed key Key-size is 128 bits long, proposed technique is easy to understand as well as easy in implantation with no hidden features. As we know that higher key length makes higher security of information so here I choose a higher key length. Some other feature of proposed encryption/decryption algorithm listed below:

- Proposed Algorithm enhancing the security due to dual approach one is encryption and another is hiding.
- Sequence of sub key selection is also providing security level.
- Proposed encryption/decryption algorithm used basic but very strong logical operation which provides higher security with higher efficiency.

Section III – Results

3.1 Result Analysis: For experiment, proposed system used desktop machine which. Configuration of that machine is Intel Pentium Dual Core T4400 1.18 Ghz, 1 GB of RAM and Window-XP SP2, in which performance data is collected. Calculated results are environment dependent so they can vary. In the experiments, the system encrypts/decrypt a various size of text file individually and encryption/decryption within hiding and un-hiding. There are some parameters are calculating by the proposed system as well as exiting system [1] which is shown in table 3.1 to 3.11. In this proposed system has run an n cycle (that is, the number of the evaluated plaintexts). In each cycle, same plaintexts are respectively encrypted by existing algorithm defined in [1] and **"Proposed Algorithm"** by copying them. Finally, the outputs of the comparison system execution time, Throughput, CPU uses and RAM uses, and measured in numeric form. Hear proposed system is developed in .Net programming language. In this proposed system comparing factors are execution time in terms of encryption and decryption time, throughput, CPU uses, RAM uses. This section presents the results of evaluating the efficiency of the proposed technique that is based on selected parameters. The execution time [] is considered the time that an encryption algorithm takes to produce a cipher text from a plaintext. Execution time is used to calculate the throughput of an encryption technique which indicates to the speed of encryption process. The throughput of the encryption process is calculated as the total plaintext in bytes encrypted divided by the execution time. The CPU process time is the time that a CPU is committed only to the particular process of calculations, which reflects to the load of the CPU. The more CPU utilization time is used in the encryption process, the higher is the load of the CPU []. The memory deals with the amount of memory space it takes for the whole process of encryption and decryption. Basically proposed system is performing encryption and decryption with hiding on several size files using same key.

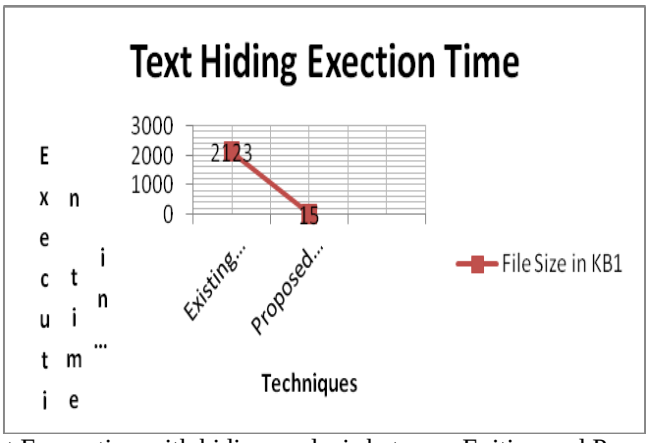
Table 3.1: Showing Comparisons between Existing and Proposed Algorithm

Tech.	File Size	Execution Time	Throughput	CPU	Memory	Mode
Existing Algo	1	0:0:20:500	0.05	50	455	Encrypt
Existing Algo	1	0:0:20:453	0.05	50	455	Decrypt
Existing Algo	2	0:0:21:375	0.0952381	50	455	Encrypt
Existing Algo	2	0:0:21:421	0.0952381	50	455	Decrypt
Existing Algo	3	0:0:23:187	0.1304348	50	455	Encrypt
Existing Algo	3	0:0:23:156	0.1304348	50	455	Decrypt
Existing Algo	4	0:0:24:843	0.1666667	50	455	Encrypt
Existing Algo	4	0:0:24:796	0.1666667	50	455	Decrypt
Proposed Algo	1	0:0:0:78	Infinity	20	443	Encrypt
Proposed Algo	1	0:0:0:15	Infinity	50	443	Decrypt
Proposed Algo	2	0:0:0:62	Infinity	50	443	Encrypt
Proposed Algo	2	0:0:0:62	Infinity	50	443	Decrypt
Proposed Algo	3	0:0:0:140	Infinity	50	443	Encrypt
Proposed Algo	3	0:0:0:156	Infinity	50	443	Decrypt
Proposed Algo	4	0:0:0:281	Infinity	50	443	Encrypt
Proposed Algo	4	0:0:0:296	Infinity	50	443	Decrypt
Proposed Algo	1	0:0:0:15	Infinity	50	446	Encrypt+Hide
Proposed Algo	1	0:0:0:0	Infinity	12	447	Unhide+Decrypt
Existing Algo	1	0:0:21:234	0.04761905	50	447	Encrypt+Hide
Existing Algo	1	0:0:21:421	0.04761905	50	446	Unhide+Decrypt

3.2.1 Encryption/Decryption with Hiding Time: - Here, **"The Proposed Algorithm (PA)"** and existing algorithm have been implemented on a number of data files varying types of content and sizes of a wide range. Encryption/Decryption time of Various Text files comparisons shown in table 3.2-3.3.

Table 3.2: Comparison of Text Encryption with hiding between Exiting and Proposed Algorithm

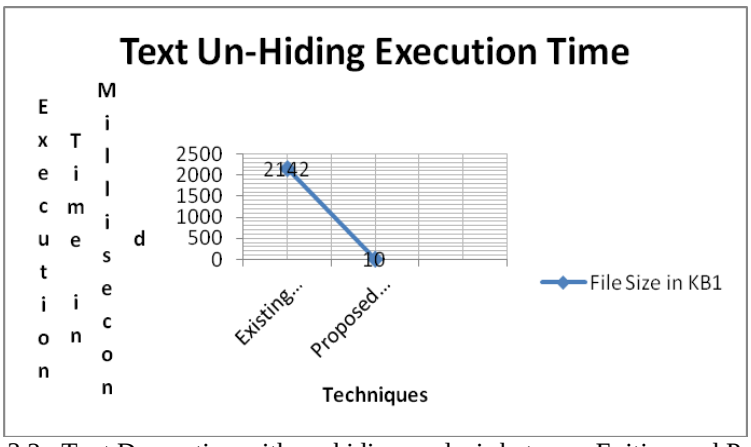
S.NO	FILE SIZE	Cover File Size	Existing Algorithm	Proposed Algorithm
	in KB		Execution Encryption Time (Approx)	
1	1	3	0:0:21:234	0:0:0:15



Graph 3.1: Text Encryption with hiding analysis between Exiting and Proposed Algorithm

Table 3.3: Comparison of Text Decryption with un-hiding between Exiting and Proposed Algorithm

S.NO	FILE SIZE	Cover File Size	Existing Algorithm	Proposed Algorithm
	in KB		Execution Decryption Time (Approx)	
1	1	3	0:0:21:421	0:0:0:10

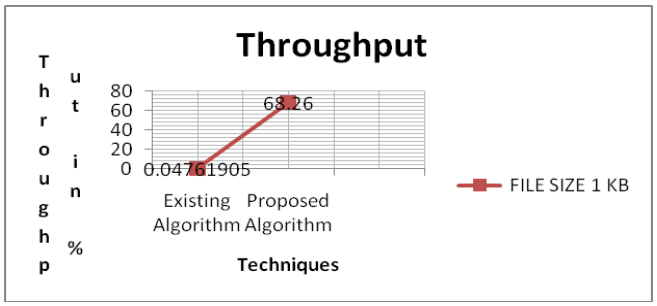


Graph 3.2: Text Decryption with un-hiding analysis between Exiting and Proposed Algorithm

3.2.2 Throughput with Hiding: - Here, **"The Proposed Algorithm"** and existing algorithm have been implemented on a number of data files varying types of content and sizes of a wide range. Throughput of Various Text files comparisons shown in table 3.4

Table 3.4: Comparison of Throughput during Encryption with Hiding between Exiting and Proposed Algorithm

S.NO	FILE SIZE	Existing Algorithm	Proposed Algorithm
	in KB	Throughput(Approx)	
1	1	0.04761905	68.26

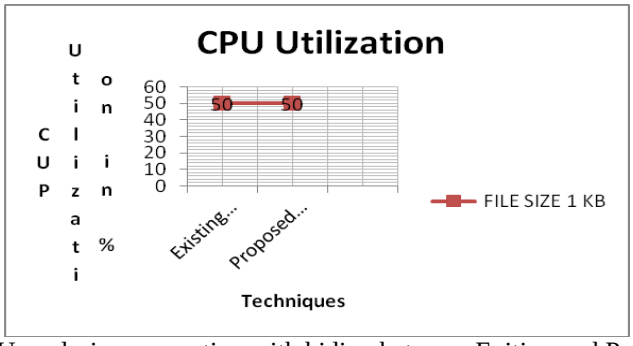


Graph 3.3: Analysis of Throughput during Encryption with Hiding between Exiting and Proposed Algorithm

3.2.3 CPU Uses with Hiding: - Here, **"The Proposed Algorithm (PA)"** and existing algorithm have been implemented on a number of data files varying types of content and sizes of a wide range. CPU Uses of Various Text files comparisons shown in table 3.5

Table 3.5: Comparison of CPU Uses during encryption with hiding between Exiting and Proposed Algorithm

S.NO	FILE SIZE	Existing Algorithm	Proposed Algorithm
	in KB	CPU Consumption in % (Approx)	
1	1	50	50

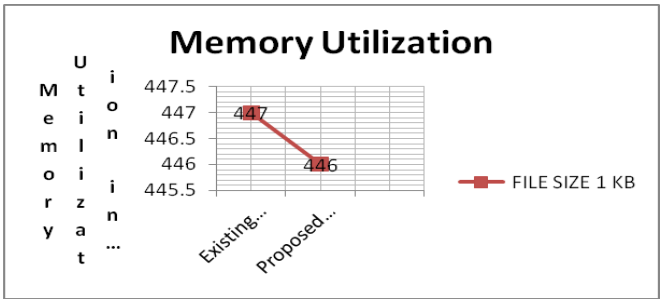


Graph 3.4: CPU Uses during encryption with hiding between Exiting and Proposed Algorithm

3.2.4 Ram Uses with Hiding: - Here, **"The Proposed Algorithm (PA)"** and existing algorithm have been implemented on a number of data files varying types of content and sizes of a wide range. RAM Uses of Various Text files comparisons shown in table 3.6

Table 3.6: Comparison of RAM Uses during encryption with hiding between Exiting and Proposed Algorithm

S.NO	FILE SIZE	Existing Algorithm	Proposed Algorithm
	in KB	RAM Uses in Bytes (Approx)	
1	1	447	446



Graph 3.5: RAM Uses during encryption with hiding between Exiting and Proposed Algorithm

Summary: From the results analysis it can clearly seen that the proposed algorithm has quite better results than existing algorithms defined in [1] and hence can be incorporated in the process of hiding of any plain text behind a cover file. However it is also clear from table 3.1 to 3.6 and graphs 3.1 to 3.5 that, by applying proposed algorithm to the files of different sizes highly security is obtain as compare to different other encryption algorithm. In execution time, throughput, CPU uses and RAM Uses the proposed algorithm have quite good results as compared to different other encryption algorithm. Table 3.2 and graph 3.1 showing the execution time of hiding a file of 1 kb in size behind a cover file of 3 kb in 1 kb file are executing through proposed algorithm then there is 15 millisecond time taken to execute where exiting

algorithm [1] are taking 2123 millisecond. Similarly at the time of un-hiding file there is 10 millisecond times taken by proposed algorithm where existing algorithm taking 2142 millisecond which is shown in table 3.3 and graph 3.2. For throughput calculation on same file size is 68.22 through proposed algorithm where 0.047 through existing algorithm which is shown in table 3.4 and graph 3.3. Similarly CPU utilization is same 50% in both algorithm which is shown in table 3.5 with graph 3.4 and Memory Utilization is almost same 447 KB in both algorithm are shown in table 3.6 with graph 3.5

Section IV – Conclusion

A new simple encryption technique has been design and developed, which is targeted for use inside of a small organization such as a college campus for faculty and staff' daily use of sending confidential files and sensitive information such that the information can be encrypted and is sent through e-mail while the encryption key is predefine because symmetric in nature which can be sent through another e-mail or via any secure communication channel. The proposed encryption technique developed and described in this research work might not be comparable to well-known and well defined earlier encryption technique but its simplicity and availability proves that proposed system can be design and developed that fit the needs of an organization without resorting to purchasing too much expensive software from the market. This research aimed at Encryption and Decryption for secure communication in public network has satisfied the goal. The development and implementation of proposed technique has given us a great satisfaction. Through lots of efforts, it has incorporated into the system several features such as authentication and integrity.

Proposed research work have described new techniques for encryption/decryption data in un-trusted server or public network and provided proof of security for the resulting crypto systems. Proposed techniques have a number of advantages are as follow:

- Proposed Technique are provably secure;
- Proposed Technique support controlled and hidden search and query isolation;
- Proposed Technique are simple and fast (More specifically, for a document of length , the encryption and search algorithms only need O(n) block cipher operations);
- Proposed Technique introduces almost no space and communication overhead.
- It conclude that this provides a powerful new building block for the construction of secure services in the un-trusted infrastructure

References

[1] Rishav Ray, Jeeyan Sanyal, Debanjan Das, Asoke Nath "A new Challenge of hiding any encrypted secret message inside any Text/ASCII file or in MS word file: RJDA Algorithm" IEEE International Conference on Communication Systems and Network Technologies 2012

[2] Dripto Chatterjee, Joyshree Nath, Suvadeep Dasgupta, Asoke Nath "A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm" published in 2011 International Conference on Communication Systems and Network Technologies, 978-0-7695-4437-3/11 $26.00 © 2011 IEEE.

[3] Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath "Advanced Steganographic Approach for Hiding Encrypted Secret Message in LSB, LSB+1, LSB+2 and LSB+3 Bits in Non standard Cover Files" International Journal of Computer Applications (0975 – 8887) Volume 14– No.7, February 2011.

[4] Debanjan Das, Megholova Mukherjee, Neha Choudhary, Asoke Nath and Joyshree Nath "An Integrated Symmetric key Cryptography Algorithm using Generalised modified Vernam Cipher method and DJSA method: DJMNA symmetric key algorithm" 978-1-4673-0126-8/11/$26.00 c_2011 IEEE.

[5] Neeraj Khanna, Joyshree Nath, Joel James, Amlan Chakrabarti, Sayantan Chakraborty and Asoke Nath " New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm" IEEE International Conference on Communication Systems and Network Technologies 2011.

[6] Diaa Salama Abdul Minaam, Hatem M. Abdual-Kader, and Mohiy Mohamed Hadhoud "Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types" International Journal of Network Security, Vol.11, No.2, PP.78-87, Sept. 2010.

[7] Akhil Kaushik, Manoj Bamela and AnantKumar "Block Encryption Standard for Transfer of Data" IEEE International Conference on Networking and Information Technology 2010.

[8] R. Venkateswaran Dr. V. Sundaram "Information Security: Text Encryption and Decryption with poly substitution method and combining the features of Cryptography" International Journal of Computer Applications (0975 – 8887)Volume 3 – No.7, June 2010

[9] Yan Wang and Ming Hu "Timing evaluation of the known cryptographic algorithms "2009 International Conference on Computational Intelligence and Security 978-0-7695-3931-7/09 $26.00 © 2009 IEEE DOI 10.1109/CIS.2009.81

[10] Md. Nazrul Islam, Md. Monir Hossain Mia, Muhammad F. I. Chowdhury, M.A. Matin "Effect of Security Increment to Symmetric Data Encryption through AES Methodology" Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing 978-0-7695-3263-9/08 DOI 10.1109/SNPD.2008.101 IEEE 2008.

[11] Nadeem.A, "A performance comparison of data encryption algorithms," IEEE Information and CommunicationTechnologies, 2006

