New Text Encryption Method Based on Hidden Encrypted Symmetric Key

Seddeq E. Ghrare, Haneen A. Barghi, Nora R. Madi

Department of Electrical and Computer Engineering, Faculty of Engineering, University of Gharyan,
P.O Box 64418 Gharyan - Libya
email: eng.ukm@gmail.com, seddeq@jgu.edu.ly

Abstract: **Cryptography is classified into two main categories which are Symmetric Key Cryptography and Asymmetric Key Cryptography. In both categories, the security level provided by any cryptographic algorithm depends on its encryption and decryption keys. In this paper a new encryption and decryption algorithm based on Hidden Encrypted Symmetric Key (HESK) is designed and implemented. The strength of this algorithm is represented in the key used for encryption and decryption process. The key itself is encrypted prior to be used for plain text encryption and cipher text decryption processes, then it is hidden inside the cipher text. The aim of hiding the key is to overcome the problem of distributing the secret key and to make the proposed algorithm more secure and difficult or even impossible to be broken. The proposed algorithm is tested on a set of plain texts of various sizes. The experimental result has been demonstrated that it is difficult to factorize the used key. The main two advantages of the proposed method are represented in the computation simplicity and security efficiency.**

Keywords: ***cryptography; encryption; decryption; plaintext; ciphertext***

I. INTRODUCTION

On the Internet, information passes from one computer to another through numerous systems before it reaches its destination. Some information, such as banking, electronic payment and electronic voting are very sensitive, therefore it should run and exchanged over the network in a robust manner and safely [1]. Cryptography is considered one of the most used ways to protect the sensitive information and prevent unauthorized people from altering that information.

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. While cryptography is the science of securing data, cryptanalysis is the science of analyzing and breaking secure communication. Cryptanalysts are also called attackers. Cryptology embraces both cryptography and cryptanalysis [4].

The history of cryptography can be broadly divided into three phases:

1. From ancient civilizations to the nineteenth century and the first part of the twentieth century, with relatively simple algorithms that was designed and implemented by hand.
2. Extensive use of encrypting electro-mechanical machines, around the period of the Second World War.
3. Ever more pervasive use of computers, about in the last fifty years, supported by solid Mathematical basis.

Cryptography was already used in ancient times, essentially in three kinds of contexts:

a) Private communications
b) Art and religion
c) Military and diplomatic use

A cryptographic algorithm is a function used for both encryption and decryption processes. This function is dependent on a key value necessary for both encryption and decryption [2]. The problem associated with the cryptographic algorithms is the security that can be provided. The strength of any cryptographic algorithm depends on the strength of the keys used. In other words, the problem of low level security of any algorithm arises from the weak encryption and decryption keys that have been used and because of the rapid growth in factorization algorithms; weak encryption and decryption keys were easily factored and discovered. To overcome this problem and to provide a good level security, the used keys should be powerful enough [3].

Based on the used keys, cryptographic algorithms can be classified into two main categories which are asymmetric key cryptographic algorithms and symmetric key cryptographic algorithms [4].

In the first category algorithms, the key used for decryption process is different from the one used for encryption process. It is extremely difficult to determine one key by analyzing the other. This allows for the free distribution of one key (i.e., public), while the key used for decryption is kept private [3,4].

The opposite is true for the second category algorithms, keys used for encryption and decryption processes are the same. This requires that sender and receiver agree on the key prior to any information exchange [3,4]. Both asymmetric and symmetric key cryptography are illustrated in Figs. 1 and 2.

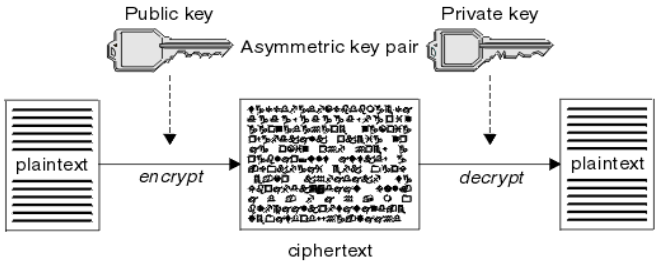

Fig. 1. Asymmetric Key Cryptography.

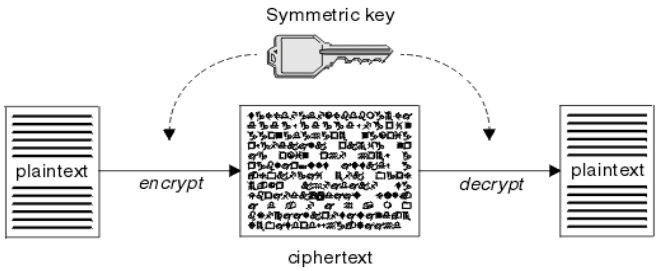

Fig. 2. Symmetric Key Cryptography.

In this paper a new encryption and decryption method based on Hidden Encrypted Symmetric Key (HESK) is designed and implemented. This algorithm starts with reading the plaintext. Then it generates the encryption and decryption key from the plain text. The key itself is encrypted prior to be used for plain text encryption and cipher text decryption processes. Then the encrypted key is hidden in the cipher text. Finally, both the encrypted key and cipher text are sent. The aim of hiding the key in the cipher text is to overcome the problem of distributing the secret key and to make the proposed algorithm more secure and difficult or even impossible to be broken. The proposed method is tested on a set of plain texts of various sizes. The experimental result has been demonstrated that it is difficult to factorize the used key. The main two advantages of the proposed method are represented in the computation simplicity and security efficiency.

II. METHDOLOGY

The cryptosystem of the proposed algorithm has been divided into three modules, as indicated in Figure 3, which are:

- Key generation module
- Data encryption module
- Data decryption module

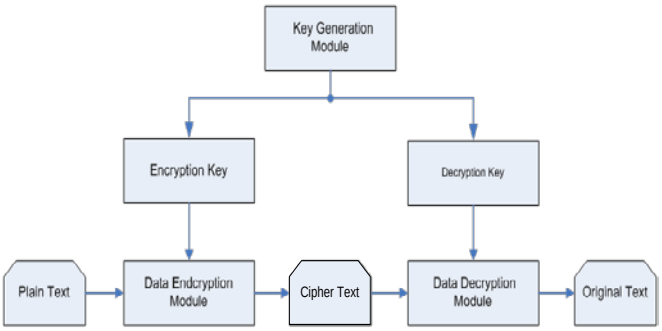

Fig. 3. Cryptosystem of the Proposed Algorithms.

A. Key Generation Module

This stage involves the generation of two keys, which could be used to encrypt the input data or message to be transferred and to decrypt the received encrypted message at the destination; those two keys are called key1 and key2 respectively. Only one of these keys can be used. In order to increase the security level of the proposed method, the chosen encryption key is encrypted and then used to encrypt the plain text in the following next stage (module).

B. Data Encryption Module

In this module any data or plain text to be sent to the receiver is encrypted prior to being transferred using the generated and encrypted keys; Key1 or Key2. The used encryption key is then inserted and hidden in the cipher text. Finally, the cipher text, which contains on the encrypted encryption keys, is sent to the destination.

C. Data Decyption Module

When the encrypted data ***(Cipher text)*** reaches the receiver, it cannot be read. In order to be read, the hidden encrypted decryption key should be extracted from the cipher text. Then the cipher text is decrypted and converted to its original form ***(plain text)*** using the extracted key.

The main steps of the proposed algorithm are as follows:
Step 1: Read the plain text
Step 2: Divide the plain text into two halves
Step 3: Generat the encryption and decryption keys
3.1: Key1 = LH (Lower half of the plaintext)
3.2: Key2 = UH (Upper half of the plaintext)
Step 4: Encrypt the encryption and decryption key
Step 5: Use the resulted encrypted key to encrypt the whole plain text
Step 6: Hide the encryption key in the ecrypted text (Cipher text)
Step7 : Send the cipher text with the hidden key to the intended reciever.

In the previous third step, which involves the key generation, there are two generated keys (key1 and key2) which means that either one can be used. The size of the used key is equal to the half of the plain text size.
The generated key is firstly encrypted, then the encrypted key is used to encrypt the plain text,and finally the encrypted key is hidded inside the cipher text.

At the reciver side the following steps should be followed in order to get the original plain text.
Step 8: Extract the hiden key from the cipher text
Step 9: Decrypt the cipher text using the extracted key
Step 10: Decrypt the key
The encryption and decryption processes are performed using the following equations:

$$C = E(k,p)$$
$$E(k,p) = (p+k) \bmod 26 \quad (1)$$

where :
C is the cipher text
K is the encryption key
P is the plain text
E is the encryption algorithm performed to encrypt the plain text (p) using the encryption key (k)

$P = D(k,C)$

$D(k,C) = (p - k) \bmod 26$ (2)

where :

C is the cipher text
K is the decryption key
P is the plain text
D is the decryption algorithm performed to decrypt the cipher text (c) using the decryption key (k)

The above steps of the proposed algorithm are implemented using Java programming language [5] and tested on a set of text file of different sizes Figure 4 shows the block diagram of the proposed algorithm.

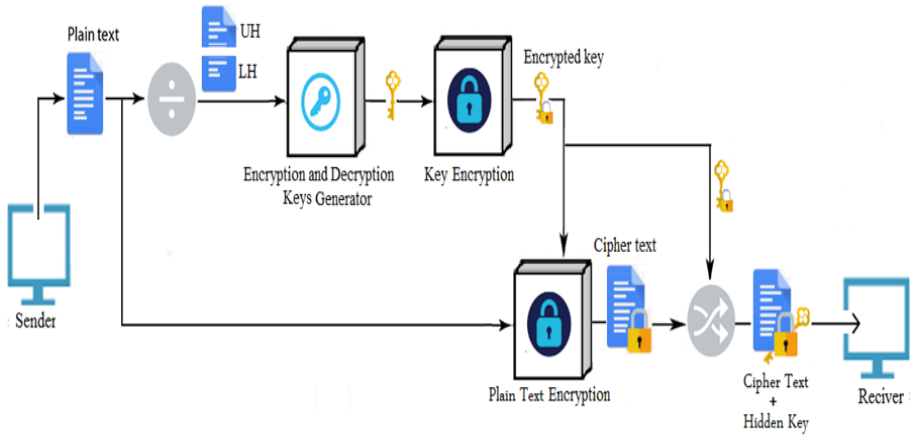

Fig. 4. Block Diagram of the Proposed Algorithm.

III. Results Presentation

As it was mentioned, the proposed algorithm is implemented using Java programming language. The results obtained by performing the proposed algorithm on the same files of sizes and executed using hardware with the following specifications:

- Windows 7 Ultimate Operating System
- Intel Core3 Processor
- CPU speed of 2.10 GHz
- RAM size of 2 GB
- HDD of 500 GB

The size of the used data files are 1KB, 10KB, 100KB, 300kB and 0.5 MB. The Graphical User Interface (GUI); which is generated by the designed program; is shown in Figure 5.

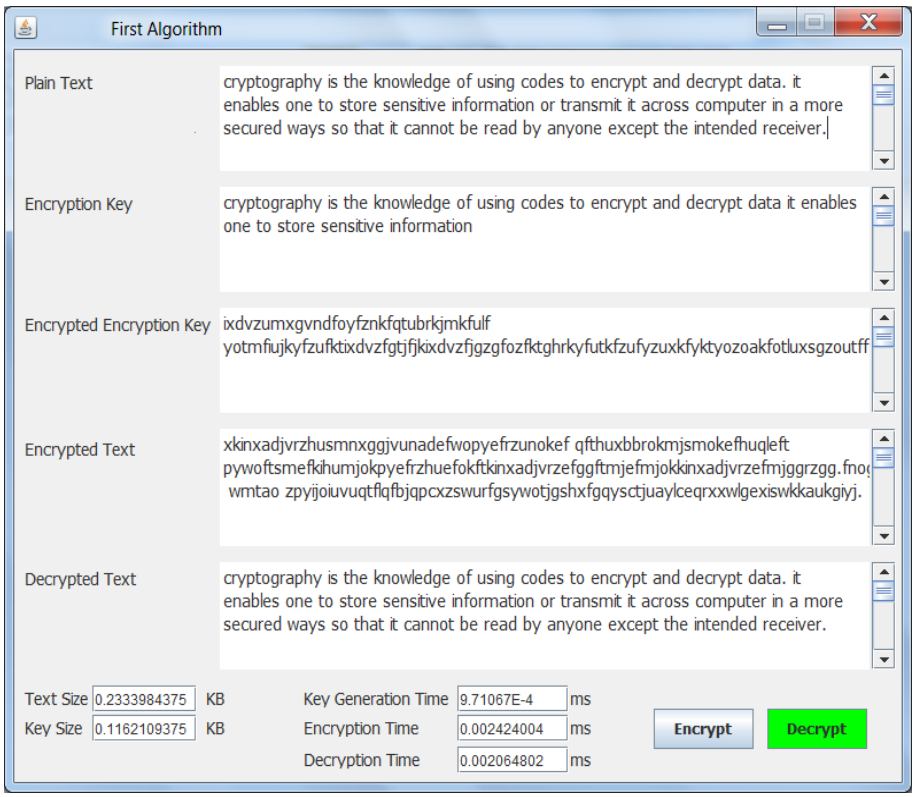

Fig. 5. The GUI of the Proposed Algorithm.

Table 1 shows the time taken for the key generation process, encryption process, and decryption process respectively. Note that when the size of the plain text is increased, the size of the generated encryption key will be longer and will take longer time. Consequently, the generated key will be stronger and harder to be broken or discovered. Moreover, the generated encryption key is encrypted before it was used for encrypting the plain text, then it is inserted into the cipher text to be hidden. So, all those steps were taken in the account and included in the time calculations.

TABLE 1. EXECUTION TIME OF THE PROPOSED ALGORITHM

Plain Text Size(KB)	Key Generation Time (Sec)	Encryption Time (Sec)	Decryption Time (Sec)
1	0.80	1.20	1.70
10	1.45	2.90	4.54
100	5.58	13.87	19.67
300	12.62	27.50	31.90
500	23.35	64.95	72.10

From the above table, the following comments can be extracted:

The overall computation time taken by the proposed algorithm is 1.20 sec, 3.00 sec, 13.00 sec, 24.00 sec, and 53.00 sec, to encrypt and decrypt plain texts of 1KB, 10KB, 100KB, 300KB, and 500KB respectively including the time taken for key generation and key encryption and hiding..

The time taken for the three stages (key generation, encryption and decryption processes) using the proposed algorithm increases whenever the size of the plain text is increased. The reason as it was explained earlier, because of that whenever the size of plain text is increased; the size of encryption key will be longer and as a result the longer time will be taken for the generation process. This can be seen from the following figure.

From figure 6, it can be noticed that the most of time is elapsed in the decryption process followed by the encryption process then key generation process. Moreover, the time is increasing and getting longer whenever the size of the plain text is increased for all the three processes.

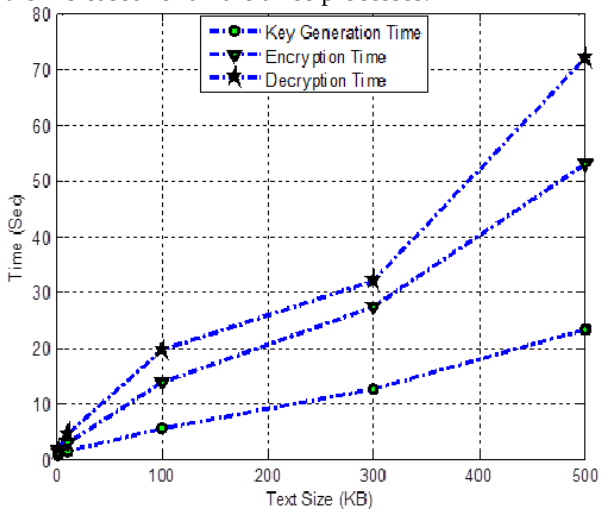

Fig. 6. Performance of the Proposed Algorithm.

IV. ANALYSIS OF THE PROPOSED ALGORITHM

For the analysis purpose, the proposed system has been implemented in JAVA programming language for demonstration intention. Proposed system has been analyzed for identification of keyword, identification of keyword distance, identification of polynomial and identification of key stream. This analysis was carried out using both Frequency analysis test and Kasiski analysis test.

A. Frequency AnalaysisTest

Frequency analysis is the study of letters or groups of letters contained in a ciphertext in an attempt to partially reveal the message. The English language (as well as most other languages) has certain letters and groups of letters appear in varying frequencies

In English language, "E" is the most common letter, appearing about 12% of the time (that is just over one in ten letters is an "E"). The next most common letter is "T" at 9%. The full frequency list is given by the graph illustrated in figure 7.

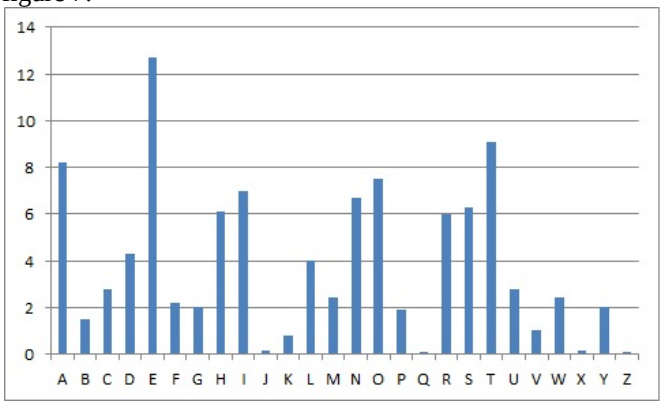

Fig. 7. Frequency of English Letters.

Further frequency analysis is applied for the ciphered text shown in Figure 10. This cipher text in resulted from the plain text enciphered by the encrypted key shown in Figure 9 using the proposed algorithm.

Cryptography is the knowledge of using codes to encrypt and encrypt data. It enables one to store sensitive information or transmit it across computer in a more secured ways so that it cannot be read by anyone excepted the intended receiver.

Fig. 8. Plain Text Sample.

ixdvzumxgvndfoyfznkfqtubrkjmkfulf
yotmfiujkyfzufktixdvzfgtjfjkixdvzfjgzgfozfk
tghrkyfutkfzufyzuxkfyktyozoakfotluxsgzoutf
f

Fig. 9. Generated Encrypted Key.

xkinxadgvzhusmnxggjvunadefwopyefrzunokefqfthu
xbbrokmjsmokefhuqleftpywoftsmefkihumjokpyefrz
huefokftkinxadjvrzefggftmjefmjokkinxadjvrzefmjg
grzgg.fnogwmtaozpyijoiuvuqtflqfbjqpcxzswurfgsy
wotjgshxfgqysctjuaylceqrxxwlgexiswkkaukgiyj.

Fig. 10. Cipher Text with a Hidden Encryption Key.

The results obtained from applying the frequency analysis test are shown in Table 2.

TABLE 2. COMPARISON RESULTS USING FREQUENCY ANALYSIS

English alphabet	Frequency of English letters	Proposed cipher
A	8.17	3.00
B	1.49	1.30
C	2.78	1.30
D	4.25	1.75
E	12.70	5.70
F	2.23	8.80
G	2.02	6.55
H	6.09	2.60
I	6.97	3.50
J	0.15	5.25
K	0.77	4.80
L	4.03	1.75
M	2.41	3.90
N	6.75	3.00
O	7.51	5.25
P	1.93	3.00
Q	0.10	2.60
R	5.99	3.00
S	6.33	3.00
T	9.06	3.90
U	2.76	4.80
V	0.98	2.20
W	2.36	2.60
X	0.15	4.80
Y	1.97	3.00
Z	0.07	3.50

From the above table it can be seen that the frequency of the letters of cipher text obtained by using the proposed algorithm is totally different from the original frequency of the English language letters which proves that the keyword of the proposed algorithm is hard to be revealed using frequency analysis.

B. Kasiski Analaysis Test

The Kasiski analysis test involves looking for strings of characters that are repeated in the ciphertext. The strings should be three characters long or more for the examination to be successful. The reason this test works is that if a repeated string occurs in the plaintext, and the distance (period) between corresponding characters is a multiple of the keyword length, the keyword letters will line up in the same way with both occurrences of the string. Then, the distances between consecutive occurrences of the strings are likely to be multiples of the length of the keyword. Thus finding more repeated strings narrows down the possible lengths of the keyword.

The Kasiski test has been applied to the text which is enciphered using the proposed algorithm. The most repeated strings of character and their distances are listed in Table

TABLE 3. REPEATED STRINGS AND PERIODS

Sequence	zhu	ywo	zef	xad	ggf
Distance	78	113	21	94	27

From the above table it can be seen that there is no any relation between the distances (periods) of those strings, which means the distances between consecutive occurrences of the strings are not multiples and as a results the keyword is hard to be revealed.

V. CONCLUSION

In this paper, text encryption and decryption algorithm based on Hidden Encrypted Symmetric Key (HESK) is designed and implemented. The strength of this algorithm is represented in the key used for encryption and decryption process. The key itself is encrypted prior to be used for plain text encryption and cipher text decryption processes, then it is hidden inside the cipher text in such a way which makes it cannot be recovered. The aim of hiding the key is to overcome the problem of distributing the secret key and to make the proposed algorithms more secure and difficult or even impossible to be broken.

The proposed algorithm was tested on a set of plain texts of various sizes. The experimental result has been demonstrated that it is difficult to factorize the used key. The main two advantages of the proposed algorithm are represented in the computation simplicity and security efficiency.

REFERENCES

[1] William Stalling, "*Cryptography and Network Security Principal and Practice*", Third Edition, Pearson (2006).

[2] Ayushi, "*A Symmetric Key Cryptographic Algorithm*", International Journal of Computer Applications (0975 - 8887), Vol. 1, No. 15, (2010)

[3] Arjen K. Lenstra and Eric R. Verheul., "*Selecting cryptographic key sizes"*. In Public Key Cryptography, pp 446-465. (2000).

[4] Prashant Kumar Arya et al , "*Comparative Study of Asymmetric Key Cryptographic Algorithms*", International Journal of Computer Science & Communication Networks,Vol 5(1),17-21 (2015)

Deitel, H.M. and P.J. Deitel., "*Java: How to Program"*, Fifth Edition, Perentice Hall Inc., Upper Saddle River, New Jersey. (2002)