Using Highly Secure Data Encryption Method for Text File Cryptography

Mua'ad M. Abu-Faraj[1†] **Ziad A. Alqadi [2††]**

The University of Jordan Albalaqa Applied University

Summary
Many standard methods are used for secret text files and secrete short messages cryptography, these methods are efficient when the text to be encrypted is small, and the efficiency will rapidly decrease when increasing the text size, also these methods sometimes have a low level of security, this level will depend on the PK length and sometimes it may be hacked. In this paper, a new method will be introduced to improve the data protection level by using a changeable secrete speech file to generate PK. Highly Secure Data Encryption (HSDE) method will be implemented and tested for data quality levels to ensure that the HSDE destroys the data in the encryption phase, and recover the original data in the decryption phase. Some standard methods of data cryptography will be implemented; comparisons will be done to justify the enhancements provided by the proposed method.
Keywords:
Cryptography, HSDE, PK, MSE, PSNR, speech file, throughput, text file, short message.

1. Introduction

The short text message is a set of letters and numbers with a small size, which does not exceed four kilobytes. As for text files, they are a set of symbols organized in a file and the size we will consider in this research paper is greater than four kilobytes. Messages and text files are widely circulated through various social media, and some of this data requires protection from intruders or parties not related to the data, as this data is confidential or of a personal nature. The process of data cryptography is one of the important processes used to protect confidential data and prevent data penetration to understand its content [1-2]. Cryptography means data encryption by destroying the original data and making it incomprehensible to anyone trying to spy on the data, the encrypted data must be recovered by applying decryption and the recovered data must match the original data [3-6].

In [7-9], Symmetric methods of data cryptography use a secret private key (PK) that is known by the sender and receiver, where this key enters the implementation of all operations related to the completion of data encryption and decryption. PK can be selected to encrypt-decrypt any secret data including short messages and text files as shown in figures 1 and 2 [10-12].

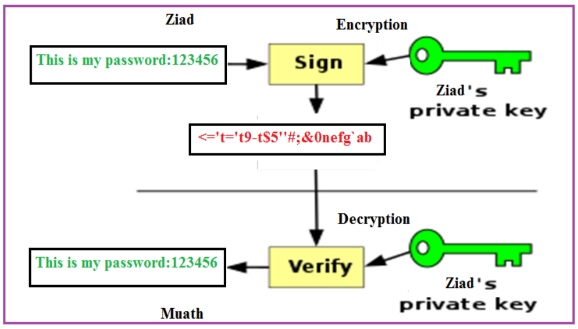


Fig. 1: Short Message Cryptography

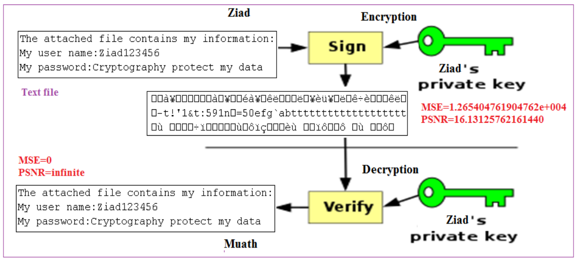


Fig. 2: Text File Cryptography

Data cryptography method is considered as a good method if it meets the following requirements [13-18]:

1) It should be easy and doable.
2) The private key must be complex and impenetrable to raise the level of security and protection of the data [4], [6].
3) It must be effective by reducing the encryption time and decoding time to the least possible, which leads to raising the permeability of the method, which is measured by the number of bytes processed per second [19-20].
4) The method should be flexible and usable to encrypt and decrypt all types of data, including large text files.
5) The method works to completely destroy the data when encrypting and return the original data when decrypting. The data quality here can be measured

Manuscript received December 5, 2021
Manuscript revised December 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.12.8

by mean squire error (MSE) and/or peak to signal ratio (PSNR) (see Eq. 1 and Eq. 2) [21-24].

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} (X(i,j) - Y(i,j))^2 \quad (1)$$

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (2)$$

The MSE value must be very high when encrypting the data, but the value PSNR must be very low, and this indicates the amount of damage caused to the data until it becomes incomprehensible and useless for any third party that has nothing to do with the data when decrypting the data, the value of MSE must very closed to zero, while the value of PSNR must be closed to infinite [25-27].

This research aims to introduce, implement and test a Highly Secured Data Encryption (HSDE) method for encrypting and decrypting short messages and text files. HSDE will be tested against DES, Triple-DES, AES, and Blowfish. The outline of this paper is divided into five sections. Firstly, we introduce the related work to different symmetric standards and a comparison between them. Secondly, we present HSDE, a method to provide a high level of data security and protection by using a complex PK. While the third section provides the implementation and the experimental results. The fourth section provides results and analysis. The final section demonstrates conclusions and future work.

2. Related Work

Many symmetric standards are now widely used in the process of short messages and text files cryptography, these methods give excellent quality parameters (MSE and PSNR) during the encryption and decryption phases, these methods vary in efficiency and it drops rapidly when the text file size increase. Some of the methods are based on data encryption standard (DES) [13], [28-31], and Triple-DES (3DES) [5], other are based on advance encryption standard (AES) [13], [28], [30], [35-36]. These methods were improved by the introduced blowfish (BF) method [37-40].

DES encrypts and decrypts data in 64-bit blocks, using a 56-bit key. It takes a 64-bit block of plaintext as input and outputs a 64-bit block of ciphertext. Since it always operates on blocks of equal size and it uses both permutations and substitutions in the algorithm. DES has 16 rounds, meaning the main algorithm is repeated 16 times to produce the ciphertext. It has been found that the number of rounds is exponentially proportional to the amount of time required to find a key using a brute-force attack. So as the number of rounds increases, the security of the algorithm increases exponentially [28-31].

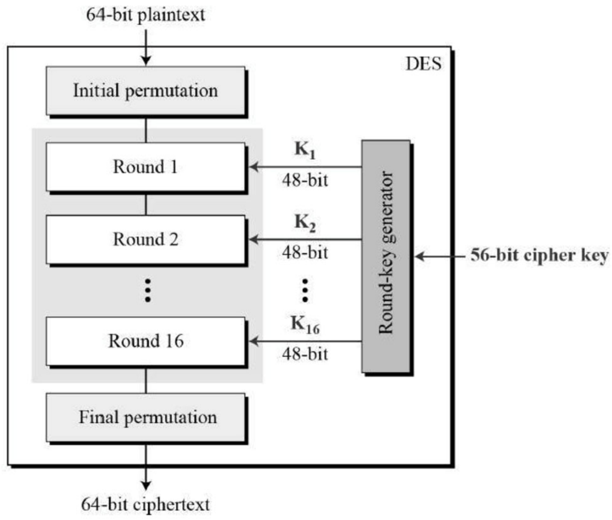


Fig. 3: Data Encryption Standard

Triple-DES is a variation of Data Encryption Standard (DES). It uses a 64-bit key consisting of 56 effective key bits and 8 parity bits. The size of the block for Triple-DES is 8 bytes. Triple-DES encrypts the data in 8-byte chunks. The idea behind Triple-DES is to improve the security of DES by applying DES encryption three times using three different keys. Triple-DES algorithm is very secure (major banks use it to protect valuable transactions), but it is also very slow [32-34]. Triple-DES encrypts data three times and uses a different key for at least one of the three passes giving it a cumulative key size of 112-168 bits. That should produce an expected strength of something like 112-bits. Triple-DES is much stronger than (single) DES; however, it is rather slow compared to some new block ciphers. However, cryptographers have determined that Triple-DES is unsatisfactory as a long-term solution, and in 1997, the National Institute of Standards and Technology (NIST) solicited proposals for a cipher to replace DES entirely [5], [29-30].

The AES algorithm (also referred to as the Rijndael algorithm) is a symmetrical block cipher algorithm that uses 128,192, or 256-bit keys to transform a block of 128-bits message into 128 bits of ciphertext which is the main reason why it is strong, secure and exponentially stronger than the DES that uses 56-bit key. A substitution-permutation, or SP network, with several rounds is used by the AES algorithm to generate ciphertext. The key length used will determine the number of rounds [13], [35-36].

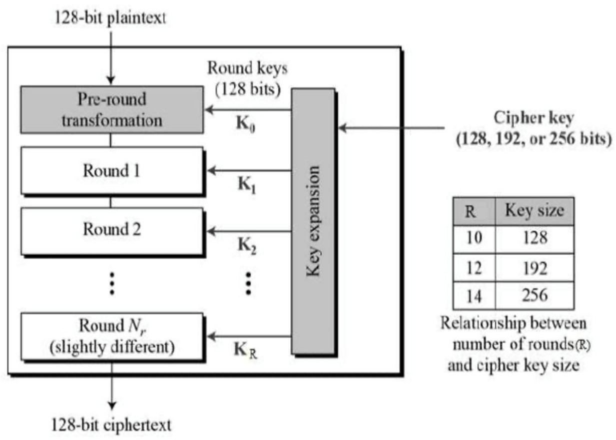


Fig. 4: Advanced Encryption Standard

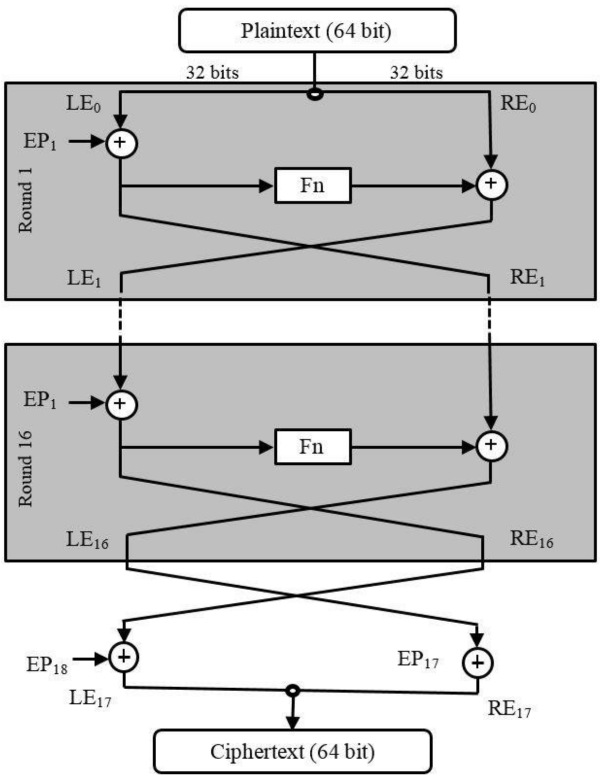


Fig. 5: Blowfish

Bruce Schneier designed Blowfish algorithm in 1994; it is symmetric encryption algorithms that uses the same secret key to both encrypt and decrypt messages. Blowfish is also a block cipher; it divides a message up into fixed length blocks during encryption and decryption [37]. The block length for Blowfish is 64-bit; messages that aren't a multiple of eight bytes in size must be padded. The Blowfish algorithm introductorily includes addition, table lookup and XOR. The table includes four S-boxes and a P-array. Blowfish is a cipher based on Feistel, and the design of the F-function used amounts to a simplification of the principles used in DES to provide the same security with greater speed and efficiency in software [37-39].

These methods are efficient in the encrypting-decrypting text file with a size up to 4 Kbytes. The standard methods used a special private key, this key is used to generate subkeys required to accomplish arithmetic and logic operations used in the encryption and decryption phase. The standard methods require performing some round to apply data cryptography, or each round a subkey must be generated from the PK, figures 3, 4, and 5 shows how these methods operate. Table 1 summarizes the mean features of these methods.

Table 1: Standard Methods Main Features

Algorithm Parameter	DES	Triple-DES	AES	Blowfish
Encryption Quality	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR
Decryption Quality	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR
Efficiency	Slow	Slow	Moderate	High
Attack	Brute force attack	Brute force attack, Known plaintext, Chosen plaintext	Side-channel attack	Dictionary attack
Structure	Feistel	Feistel	Substitution-Permutation	Feistel
Block Cipher	Binary	Binary	Binary	Binary
PK Length (bit)	56	112, 168	128, 192, 256	32-448
Block size(bit)	64	64	128	64
Rounds	16	48	10,12,14	16
Flexibility to Modification	No	Yes	Yes	Yes
Simplicity	No	No	No	No
Security level	Adequate	Adequate	Excellent	Excellent
Throughput	Low	low	Low	High

3. The HSDE Method

The speech file is one of the most common and widely used types of data, and it can be easily obtained due to the availability of many possibilities for recording and storing speech files for people. A digital speech file is a set of samples recorded at successive time intervals and the values of these samples (amplitudes) are organized into a onecolumn array (mono speech) or two-column array (stereo speech), and these values are often fractional and confined between -1 and +1 as shown in figure 6. If you would like to itemize some parts of your manuscript, please make use of the specified style "itemize" from the drop-down menu of style categories

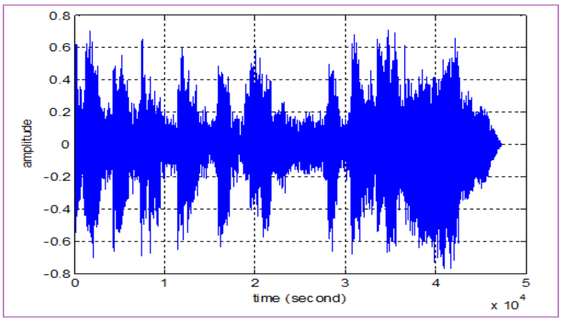


Fig. 6: Speech signal example

Highly Secured Data Encryption method uses the speech file to generate the private key to perform text file encryption and decryption process, taking into account the following:

- The audio file must be confidential and agreed upon by the sender and receiver, and it is not circulated through various social media.
- The ability to change the speech files from time to time and easily, to ensure data protection.
- The possibility of changing the size of the speech files by reducing it or increasing it to suit the size of the text to be encrypted.

The process of encrypting text files in the proposed method is implemented in two stages: the first stage is the generation of the private secret key [41], and the second stage is the encryption process.

The private key is generated by performing the following steps:

- Read the secret speech file
- Read the text file to be encrypted
- Restore the dimensions of the text file
- Resizing the speech file by converting its dimensions to dimensions equal to the dimensions of the audio file
- Converts the values in the output speech file to values identical in type to the values of the text file

Figure 7 shows an example of preparing the private key:

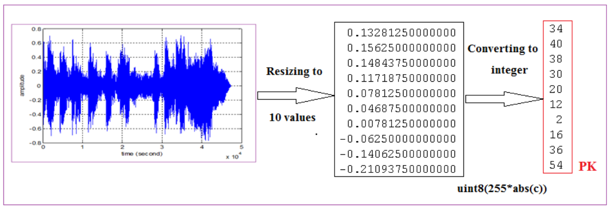


Fig. 7: PK Generation

The private key then will be used to apply XORing with the text file ASCII value to get the encrypted text file.

The decryption phase can be implemented in the same manner using the PK generation phase, the XORing the generated PK with the encrypted text file to get the decrypted text file.

Table 3 and Table 4 show the encrypted messages using two different speech files.

Table 2: Encryption results using speech 1

Message	Length	Encrypted	Encryption time (second)	MSE	PSNR
Muaad Abu Faraj	15	Íõáôè Áâõ Æáòáê	0.01194471	16782	12.6846
Ziad alqadi	11	úéáä áìñááé	0.0118441	16384	13.3886
Jordan university	17	Êïòäáî õîéö åòóéôù	0.0123511	4.0386e+003	25.1068
Albalqa Applied University	26	¾ ß¾ ß a	0.0124289	5.6939e+003	21.6719

Table 3: Encryption results using speech 2

Message	Length	Encrypted	Encryption time (second)	MSE	PSNR
Mua'ad Abu Faraj	15	!5 <t6!t5&5>	0.026000	4.00106666666667e+003	12.300315983760 54
Ziad alqadi	11	=50t58%50 =	0.024000	3.30909090909090 9e+003	14.027516013862 68
Jordan university	17	-;&05:t!:=" 1&'= -	0.026023	4.27011764705882 3e+003	12.321844333459 53
Albalqa Applied University	26	8658%5t$$ 8=10t:="1 &'= -	0.027090	4.01353846153846 1e+003	12.941525506259 19

4. Implementation and Experimental Results

A speech file was selected, the proposed method was written in Matlab code and implemented using processor i7 with 2.4 GHz processor, text files with various sizes were taken, encrypted, and decrypted, table 4 shows the obtained experimental results.

Table 4: Text files Cryptography Results

Text file size (Kbytes)	Encryption time (second)	Decryption time (second)	MSE	PSNR
1	0.119520	0.119520	2.2045e+004	10.8170
2	0.121679	0.121679	2.0917e+004	11.3421
4	0.122938	0.122938	2.1824e+004	10.9176
8	0.127936	0.127936	2.1618e+004	11.0126
16	0.132083	0.132083	2.1801e+004	10.9283
32	0.139289	0.139289	2.1619e+004	11.0121
64	0.167531	0.167531	2.1743e+004	10.9549
128	0.211526	0.211526	2.1697e+004	10.9759
256	0.281019	0.281019	2.1647e+004	10.9992
512	0.485115	0.485115	2.1678e+004	10.9846
1024	0.874237	0.874237	2.1676e+004	10.9856
Average	0.2530	0.2530		
Throughput (Kbytes per second)	735.54	735.5372		
Throughput (Mbytes per second)	0.71830	0.71830		

Several short messages were also taken and they were encrypted-decrypted, table 5 shows the obtained experimental results:

Table 5: Short messages (small text files) cryptography results

Text file size (bytes)	Encryption time (second)				
	DES	Triple-DES	AES	Blowfish	Proposed
11	0.0045762456441	0.0053179672571	0.0045176051896	0.0015239523188	0.0118441
15	0.0062403359699	0.0072517765377	0.0061603737103	0.0020781197948	0.01194471
17	0.0070729796333	0.0082186769781	0.0069817544784	0.0023551993318	0.0123511
26	0.0108165856138	0.0125697419825	0.0106779769062	0.0036020694137	0.0124289
30	0.0124806799322	0.0145035473667	0.0123207434216	0.0041562339976	0.0125197
40	0.0166408962517	0.0193380628544	0.0164276554384	0.0055416457965	0.0134685
50	0.0208011185649	0.0241725793413	0.0205345694453	0.0069270579964	0.0142243
60	0.0249613498761	0.0290070943206	0.0246414878522	0.0083124681953	0.0156277
70	0.0291215641876	0.0338416298197	0.0287483986680	0.0096978793942	0.0164734
80	0.0332817875724	0.0386761265026	0.0328553134659	0.0110832899931	0.0174552
90	0.0374420198155	0.0435106431954	0.0369622243728	0.01246870047920	0.0184489
100	0.0416022341287	0.0483451569872	0.0410691390896	0.01385411598191	0.022000
Average	0.02041981643252	0.02372941692861	0.02015810350319	0.00680006105778	0.01489887583333

The standard methods were also implemented using the same short messages, and the results are shown in Table 5. The same big text files were selected and encrypted-decrypted using each of the standard methods; the obtained experimental results are shown in Table 6:

Table 6: Text files cryptography results using standard methods

Text file size (K bytes)	Encryption time (second)				Improvements of the proposed method
	DES	Triple - DES	AES	Blowfish	
1024	45.2796	50.7307	44.4527	15.3520	Yes
512	21.1378	26.3953	21.2264	6.2450	Yes
256	10.4199	11.9827	10.1137	3.6399	Yes
128	4.9900	6.1413	5.3563	1.9007	Yes
64	2.7300	3.0707	2.2783	0.9782	Yes
32	1.3150	1.5453	1.3771	0.4623	Yes
16	0.6575	0.7627	0.6523	0.2245	Yes
8	0.3737	0.3763	0.3334	0.1269	Yes
4	0.1669	0.1882	0.1656	0.0657	No
2	0.0884	0.0981	0.0871	0.0310	No
1	0.0449	0.0465	0.0429	0.0140	No
Average	7.9276	9.2125	7.8260	2.6400	
Throughput (K bytes per second)	23.4738	20.1998	23.7785	70.4890	

5. Results Analysis

Referring to the results shown in tables 2, 3, and 4 we can see that the proposed method satisfies the quality requirement by providing excellent values or MSE and PSNR in the encryption and decryption phase.

For short messages cryptography, the proposed image has an efficiency that varies close to BF method efficiency (see table 5 and 6), figure 8 shows a comparison of encryption time for the implemented methods.

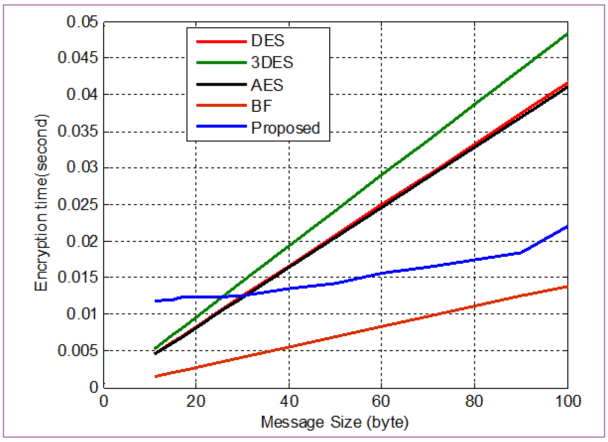


Fig. 8: Encryption Methods Time Comparison

For the standard methods, the encryption-decryption times grows rapidly when the text file size grows (see table 6), and here the proposed method will be the most efficient by keeping the encryption-decryption time minimal, which means that the proposed method gives an excellent improvement to the data cryptography throughput, this is shown in figures 9 and 10.

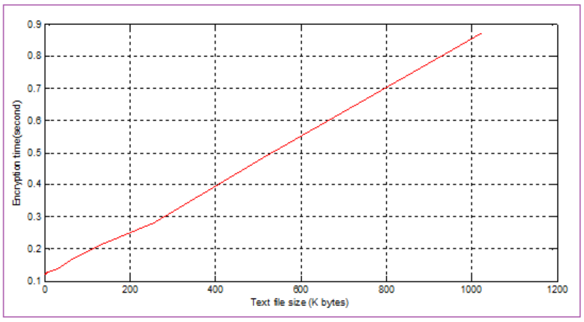


Fig. 9: Proposed Method Encryption Time (for big text files)

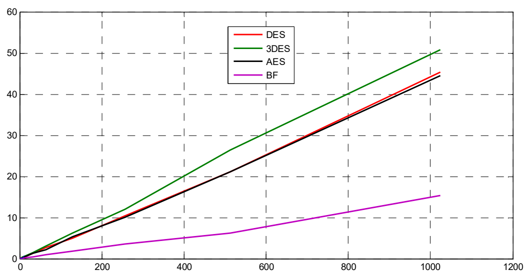


Fig. 10: Standard Methods Encryption Time (for big text files)

HSDE method provides a high-security level, here the private key is a complex one, thus the hacking process [42-44] will be impossible, the speech file which must be used to generate PK is to be kept in secrete and it can be replaced by any other speech file any time when needed, this key can be used to encrypt-decrypt any text file with any size (smaller or bigger than the speech file size).

Table 7 summarizes the main features of the proposed method compared with a standard method.

Table 7: Standard Methods and Proposed Method Main Features

Algorithm parameter	**DES**	**3DES**	**AES**	**Blowfish**	**Proposed method**
Encryption quality	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR
Decryption quality	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR
Efficiency	Slow	Slow	Moderate	High	**Excellent**
Attack	Brute force attack	Brute force attack, Known plaintext, Chosen plaintext	Side-channel attack	Dictionary attack	Impossible
Structure	Feistel	Feistel	Substitution-Permutation	Feistel	Data resizing
Block cipher	Binary	Binary	Binary	Binary	Decimal
PK length (bit)	56	112, 168	128, 192, 256	32-448	Any length
Block size (bit)	64	64	128	64	Any length
Rounds	16	48	10,12,14	16	1
Flexibility to modification	No	Yes	Yes	Yes	Yes
Simplicity	No	No	No	No	Yes
Security level	Adequate	Adequate	Excellent	Excellent	Excellent
Throughput	Low	low	Low	High	Very high

6. Conclusions

A method of short messages and text files cryptography was introduced, implemented, and tested; other standard methods (DES, 3DES, AES, and BF) were also implemented using the same messages and text files. The proposed method added enhancements to the standard methods of data cryptography by rapidly increasing the efficiency and throughput of the encryption-decryption process. The proposed method provides an excellent level of cryptography quality by keeping MSE and PSNR acceptable and meets the requirements of good cryptography. The introduced method provides a high level of data security and protection by using a complex PK, this key is to be generated by a secrete and replicable speech file making the hacking process impossible. The proposed method can be used easily to protect short messages and text files of any size. The proposed method is easy to apply and easy to modify and can be used in the future to protect digital images and audio files included in many multimedia applications.

References

[1] Burnett, S., and Paine, S. : *RSA Security's Official Guide to Cryptography*.McGraw-Hill (2001)

[2] Disina, A. H., Jamel, S. , Aamir, M., Pindar, Z. A., Deris M. M., and Mohamad, K. M.: *A key scheduling algorithm based on dynamic quasigroup string transformation and All-Or-Nothing key derivation function*. Journal of Telecommunication, Electronic and Computer Engineering, vol. 9 (3-5), pp. 1-6 (2017)

[3] Jamel, S., Deris, M. M., Yanto, I. T. R., and Herawan, T.: *The hybrid cubes encryption algorithm (HiSea)*. Communications in Computer and Information Science, Springer-Verlag Berlin Heidelberg, vol. 154, pp. 191-200 (2011)

[4] Stallings, W.: *The RC4 stream encryption algorithm*. In Cryptography and network security, 2005.

[5] Sasi, S. B., Sivanandam, N., and Emeritus, : *A survey on cryptography using optimization algorithms in WSNs*. Indian Journal of Science and Technology, vol. 8(3), pp. 216-221 (2015)

[6] Jamel, S.: *The hybrid cubes encryption algorithm (HiSea)*. Ph.D Thesis, Univ. Tun Hussein Onn Malaysia (UTHM), Johor, Malaysia, pp. 1-138 (2012)

[7] Ahmad, S., Alam, K. M. R., Rahman, H., and Tamura, S.: A *comparison between symmetric and asymmetric key encryption algorithm based decryption mixnets*. In Proceedings of the IEEE International Conference on Networking Systems and Security (2015)

[8] Fujisaki, E. and Okamoto, T.: *Secure integration of asymmetric and symmetric encryption schemes*. Journal of Cryptology, vol. 26 (1), pp. 80-101 (2013)

[9] Salama, D., Minaam, A., Abdual-kader, H. M, and Hadhoud, M. M.: *Evaluating the effects of symmetric cryptography algorithms on power consumption for different data types*. International Journal of Network Security, vol. 11(2), pp. 78-87 (2010)

[10] Goldwasser, S. and Bellare, M. Lecture Notes on Cryptography, Cambridge, Massachusetts (2008)

[11] Kaushik, A., Barnela M., and Kumar, A.: *Keyless user defined optimal security encryption*. International Journal of Computer and Electrical Engineering, vol. 4 (2), pp. 2-6 (2012)

[12] Stallings, W.: Cryptography and network security: principles and practices. Prentice Hall, (2005)

[13] Mandal, K., Parakash, C., and Tiwari, A.: *Performance evaluation of cryptographic algorithms: DES and AES*. In Proceeding of the IEEE Students' Conference on Electrical, Electronics and Computer Science: Innovation for Humanity (SCEECS), (2012)

[14] Ebrahim, M., Khan, S., and bin Khalid U.: *Symmetric algorithm survey: A comparative analysis*. International Journal of Computer Applications, vol. 61(20), pp. 12-19, (2013)

[15] Kumar, N., and P. Chaudhary, P.: *Performance evaluation of encryption/decryption mechanisms to enhance data security*. Indian Journal of Science and Technology, vol. 9(20), pp.1-10 (2016)

[16] Disina, A. H., Pindar, Z. A., and Jamel, S.: Enhanced Caeser cipher to exclude repetition and withstand frequency cryptanalysis. Journal of Network and Information Security, (2015)

[17] Palagushin, V., and Khomonenko, A. D*.: Evaluation of cryptographic primitives security based on proximity to the latin square*. In Proceeding of the IEEE 18th conference of fruct association, pp. 266-271(2016)

[18] Jamel, S. H. and Deris, M. M.: *Diffusive primitives in the design of modern cryptographic algorithms*. In proceedings of the International Conference on Computer and Communication Engineering (ICCCE08): Global Links for Human Development, pp. 707-710 (2008)

[19] Nadeem, A. and Javed, M. Y.: *A performance comparison of data encryption algorithms*. International Conference on Information and Communication Technologies, pp. 84-89, 2005

[20] S. Manku, S., and Vasanth, K.: *Blowfish encryption algorithm for information security*. ARPN Journal of Engineering and Applied Sciences, vol. 10(10), pp. 4717-4719 (2015)

[21] Daemen, J., Rijmen, V., and Leuven, K. U.: *AES Proposal: Rijndael*. (NIST), National Institute of Standards, (1999)

[22] Escala, A., Herold, G., and Ràfols, C.: *An algebraic framework for Diffie-Hellman assumptions*. Journal of Cryptology (2015)

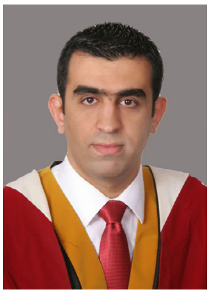
[23] Jorstad N., and Landgrave.: *Cryptographic algorithm metrics*. In 20th National Information Systems Security (1997)

[24] Mushtaq, M. F., Akram, U., Khan, I., Khan, S. N., Shahzad, A., and Ullah, A.: *Cloud computing environment and security challenges: A review*. International Journal of Advanced Computer Science and Applications, vol. 8(10), pp. 183-195 (2017)

[25] Hercigonja, Z., Gimnazija, D., and Varazdin, C.: *Comparative analysis of cryptographic algorithms and advanced cryptographic algorithms*. International Journal of Digital Technology & Economy, vol. 1(2), pp. 1-8 (2016)

[26] Stallings, W.: *Cryptography and Network Security: Principles and Practice*, 5th ed. Prentice Hall Press (2010)

[27] William, E. B., Barker, C.: *Recommendation for the triple data encryption algorithm (TDEA) block cipher*. NIST Special Publication 800-67 (2012)
[28] Maqsood, F., Ali, M. M., Ahmed, M., and Shah, M. A.: *Cryptography: A comparative analysis for modern techniques*. International Journal of Advanced Computer Science and Applications, vol. 8(6), pp. 442-448 (2017)
[29] Alshahrani, A. M., and Walker, S.: *Implement a novel symmetric block cipher algorithm*. International Journal on Cryptography and Information Security, vol. 4 (4), pp. 1-11 (2014)
[30] Smid M. E., and Branstad D. K.: *Data Encryption Standard: past and future*. In Proceedings of the IEEE, vol. 76 (5), pp. 550-559 (1988)
[31] N. I. of S. and T. NIST: *Data Encryption Standard (DES)*. Federal Information Processing Standards Publication (FIPS PUB 46-3), vol. 25(10), pp. 1-22 (1999)
[32] Dworkin, M.: *Recommendation for block cipher modes of operation*, NIST Spec. Publ. 800-38B, (2005)
[33] Patil, P., Narayankar, P., Narayan, D. G., and Meena, S. M.: *A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish*. In Procedia Computer Science, vol. 78, pp.617-624 (2016)
[34] Silva, N. B. F., Pigatto, D. F., Martins, P. S., and Branco, K. R. L. J. C.: *Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general-purpose computer*. Journal of Network and Computer Applications, vol. 60, pp. 130-143 (2016)
[35] N. I. of Standards-(NIST), Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197 (2001)
[36] Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J., and Roback, E.: *Report on the development of the advanced encryption standard (AES)*. National Institute of Standards and Technology, pp. 1-116 (2000)
[37] Nie, T., and Zhang, T.: *A study of DES and Blowfish encryption algorithm*. In Proceedings of 10th IEEE Region Annual International Conference TENCON, pp. 1-4 (2009)
[38] Schneier, B.: *Description of a new variable-length key, 64-bit block cipher (Blowfish)*. In Proceedings of the Fast Software Encryption: Cambridge Security Workshop Cambridge, U. K., pp. 191-204 (1994)
[39] Schneier, B.: *Description of a new variable-length key, 64-bit block cipher (Blowfish)*. In Proc. Fast Softw. Encryption Cambridge Security. Work. Cambridge, U. K., pp. 191-204 (1994)
[40] Mushtaq, M. F., Jamel, S. , Mohamad, K. M. , .Khalid, S. A. A., and Deris, M. M.: *Key generation technique based on triangular coordinate extraction for hybrid cubes*. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. (3-4), pp. 195-200 (2017)
[41] Disina, A. H., Jamel, S., Pindar, Z. A., and Deris M. M.: *All-or-nothing key derivation function based on quasigroup string*. In proceeding of IEEE International Conference on Information Science and Security (ICISS), pp. 6-10 (2016)
[42] Shannon, C. E.: *Communication theory of secrecy systems*. Bell System Technical Journal, vol. 28 (4), pp. 656-715 (1949)
[43] Mel, H. X., and Baker. D. M.: *Cryptography decrypted*. Addison-Wesley (2001)



Mua'ad Abu-Faraj received the B.Eng. degree in Computer Engineering from Mu'tah University, Mu'tah, Jordan, in 2004, the M.Sc. degree in Computer and Network Engineering from Sheffield Hallam University, Sheffield, UK, in 2005, and the M.Sc. and Ph.D. degrees in Computer Science and Engineering from the University of Connecticut, Storrs, Connecticut, USA, in 2012. He is, at present, an Associate Professor at The University of Jordan, Aqaba, Jordan. He is currently serving as a reviewer for the IEEE Micro, IEEE Transactions on Computers, Journal of Supercomputing, and International Journal of Computers and Their Applications (IJCA). His research interests include computer architecture, reconfigurable hardware, image processing, cryptography, and wireless networking. Dr. Abu-Faraj is a member of the IEEE, ISCA (International Society of Computers and their Applications), and JEA (Jordan Engineers Association).

Ziad A. Alqadi received the B.E., M. E., and Dr. Eng. degrees from Kiev Polytechnic Institute. in 1980, 1983, and 1986, respectively. After working as, a researcher from 1986, an assistant professor from 1991 in the department of Electrical Engineering, Amman Applied College, and an Associate Professor from 1996 in the Faculty of Engineering Technology, he has been a professor at Albalqa Applied. since 2010. His research interest includes signal processing, image processing, data security and parallel processing.