

Electronic Medical Information Encryption Using Modified Blowfish Algorithm

Noah Oluwatobi Akande[1(✉)], Christiana Oluwakemi Abikoye[2], Marion Olubunmi Adebiyi[3], Anthonia Aderonke Kayode[1], Adekanmi Adeyinka Adegun[4], and Roseline Oluwaseun Ogundokun[1]

[1] Data and Information Security Research Group, Computer Science Department, Landmark University, Omu-Aran, Kwara, Nigeria
akande.noah@lmu.edu.ng
[2] Computer Science Department, University of Ilorin, Ilorin, Kwara, Nigeria
[3] Department of Computer and Information Sciences, Covenant University, Ota, Nigeria
[4] Discipline of Computer Science, University of Kwazulu-Natal, Durban, South Africa

Abstract. Security and privacy of patients' information remains a major issue of concern among health practitioners. Therefore, measures must be put in place to ensure that unauthorized individual do not have access to this information. However, the adoption of digital alternative of retrieving and documenting medical information has further opened it up to more attacks. This article presents a modified blowfish algorithm for securing textual and graphical medical information. The F-function used in generating round sub-keys was strengthened so as to produce a strong key that could resist differential attacks. Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI) of 98.85% and 33.65% revealed that the modified algorithm is sensitive to changes in its key and also resistive to differential attacks. Furthermore, the modified algorithm demonstrated a better encryption and decryption time than the existing blowfish algorithm.

Keywords: Data and information security · Decryption · Encryption · Medical information security · Modified blowfish algorithm

1 Introduction

The advancement in Information and Communication Technologies (ICT) have caused a major shift from traditional means of medical information archiving to the electronic or digital alternative. Electronic Medical Information (EMI) entails the use of state-of-the-art ICT tools in retrieving the needed information about a patients' health condition with a view to facilitate a faster, easier, efficient and cost effective healthcare practices. While a vast majority of patients want to use digital devices to monitor their health conditions, medical practitioners also believe that EMI will facilitate effective sharing of medical information among medical practitioners and health care information technology systems without location being a barrier [14]. However, the use of these

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11623, pp. 166–179, 2019.
https://doi.org/10.1007/978-3-030-24308-1_14

ICT tools has raised security concerns about the confidentiality, integrity and availability of EMI. The content and nature of data available in health care industries have made them vulnerable to theft and data fraud [16]. A KPMG survey reported that 81% of 223 US healthcare organizations and 110 million patients in the US had their data breached in 2015 [18]. Despite the massive threats recorded by healthcare industries across the world in 2016, 95% of healthcare organizations still do not use any software for information security governance or risk management [24]. Therefore, healthcare sector faces a larger attacks and threats than other sectors perhaps for financial gains, political interest or to expose security flaws [22]. In addition, the effect of this security breach is not limited to patients' and medical personnel's psychological distress but could also result in financial loss and reputation harm [12, 28]. Though authors [20] and [32] identified Denial of Service (DoS), ransomware, malware, cryptographic attacks, privilege escalation, injection and web security exploits as some of the techniques used by attackers to infiltrate EMI, new techniques are being employed daily. Therefore, authors in [1] opined that the most effective way to secure data from attackers is to either protect the medium through which information is being sent or to put measures in place to secure the actual information being sent. To this effect, the use of biometrics, encryption, firewalls and smartcards as some security techniques that could be used to secure EMI was proposed in [3]. However, according to [30], the best approach to securing EMI is via cryptographic techniques. Leveraging on this, a modified blowfish cryptographic algorithm for encrypting and decrypting EMI is proposed in this research. Blowfish is one of the symmetric cryptographic techniques which has been widely employed to secure data. As an unpatented and license-free encryption algorithm, blowfish is known to be the fastest and simplest symmetric cryptographic algorithm [9, 19]. However, in addition to blowfish being best appropriate for instances where the key remains constant [27], it does not provide authentication and non-repudiation [10]. Blowfish key generation process depends on its Feistel function (F-Function). The default F–function uses two OR operation and one XOR operation in generating the needed key for each rounds. This article reports a modified blowfish algorithm which is aimed at further strengthening the weak keys limitation of blowfish algorithm. This was achieved by replacing the existing blowfish F-Function which uses use two XOR and one OR operation with a F-function that uses two XOR and one OR operation.

2 Existing Blowfish Algorithm

Traditional blowfish algorithm is a 64-bit symmetric block cipher that encodes an input plaintext one block at a time. As a symmetric cryptographic algorithm, the same key length that ranges from 32 bits to 448 bits is used for both encryption and decryption. The cryptographic process of blowfish algorithm majorly involves data encryption and key expansion [15, 29]. The data encryption is achieved with a 16-round Feistel network which splits the input data block into two equal halves before carrying out encryption in multiple rounds. Furthermore, the key expansion involves the use of four 32 bits S-boxes and a P-array that comprises of eighteen 32-bit sub-keys which are generated before the data encryption and decryption [2]. For the sub-key generation,

blowfish algorithm uses a F-function; this splits a 32-bit plaintext into four equal parts. Each 8-bit sub-division is then converted into a 32-bit data stream using the S-box. The resultant 32-bit data from each 8-bit sub-division is finally XORed together to produce the final 32-bits output. This shows that the F-function is an integral part of the blowfish encryption and decryption process. The F-function used for sub-key generation is shown in Fig. 1:

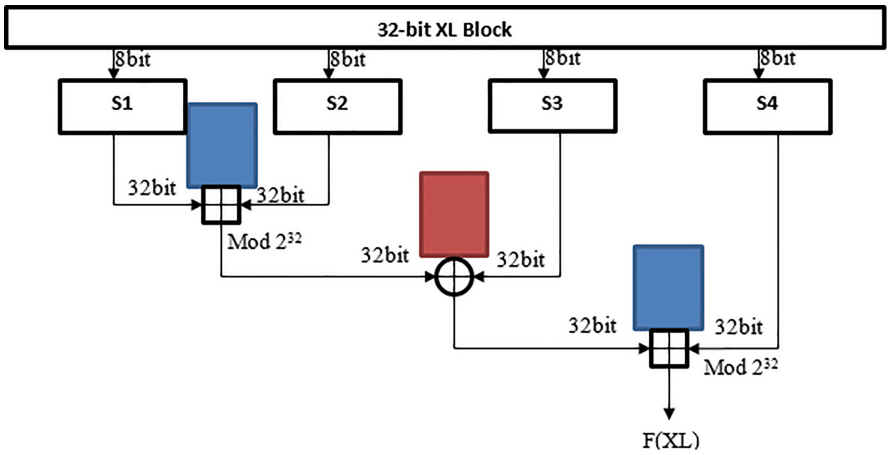

Fig. 1. Existing blowfish function F

Similarly, the algorithm below is used for the sub-keys generation:

(a) start
(b) Set the P-array to be empty then assign a fixed string to the four S-boxes. The strings are made up of hexadecimal digits: P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.
(c) Carry out a XOR operation between P1 and the first 32 bits of the key, P2 and the second 32-bits of the key until all the bits of the whole P-array has been XORed.
(d) Using subkeys generated in (c), encrypt all-zero strings with blowfish algorithm.
(e) Replace P1 and P2 with the result generated in (d).
(f) Use Blowfish algorithm to encrypt the output of (d) with the modified subkeys generated in (c)
(g) Replace P3 and P4 with the result generated in (f).
(h) repeat this process until all entries in P array has been changed.
(i) Stop

2.1 Encryption and Decryption

The main goal of blowfish algorithm is to make a message undecipherable by a third party, this is called encryption while decryption attempts to restore an encrypted

message back to its original state. The encryption and decryption process are similar except that the P-arrays (P0, P1 … P17) are used in reverse during the decryption process. However, the key used for encryption remains unchanged and are still used for the decryption process. The pseudocode for blowfish encryption is given as:

Step 1: Split the input 64bit data into xL and xR where each half contains 32-bit half data
Step 2: Repeat steps (a) to (c) while i = 1 to16

(a) Carry out XOR operation between xL and P[i].
(b) Compute the F-function of xL such that:

$$F(xL) = ((S_1 + S_2, \text{ mod } 2^{32}) \text{ xor } S_3) + S_4, \text{ mod} 2^{32}$$

(c) XOR F(xL) with xR.

Step 3: Swap xR and xL.
Step 4: Carry out XOR operation between xR and P[16].
Step 5: Carry out XOR operation between xL and P[17].
Step 6: Merge xR and xL.

The pseudocode for the decryption process is given as:

Step 1: Split the input 64bit data into xL and xR where each half contains 32-bit half data
Step 2: Repeat steps (a) to (c) while i = 16 to1

(a) Carry out XOR operation between xL and P[i].
(b) Compute the F-function of xL such that:

$$F(xL) = ((S_1 + S_2, \text{ mod} 2^{32}) \text{ xor } S_3) + S_4, \text{ mod } 2^{32}$$

(c) XOR F(xL) with xR.

Step 3: Swap xR and xL.
Step 4: Carry out XOR operation between xR and P[1]
Step 5: Carry out XOR operation between xL and P[0].
Step 6: Merge xR and xL.

3 Related Works

Several encryption techniques have been used to secure EMI which could be textual, audio as well as images. Medical image encryption using cosine number transform was proposed by Lima, Madeiro, and Sales [21]. The technique was aimed at correcting

rounding-off errors that may cause discrepancies between original and encrypted images. the key sensitivity of the technique was evaluated by attempting to decrypt the cipher with a wrong key that is slightly different from the right one. The number of pixels change rate measured revealed that the image obtained with a wrong key is completely different from the original image. The technique was also shown to be secure against entropy attack as well as known-plaintext and chosen- plaintext attacks. Also, an adaptive medical image encryption technique using an improved chaotic mapping was proposed by Chen and Hu [6]. In addition to securing medical images, the technique was aimed at overcoming the flaws of the existing chaotic image encryption algorithm. Initially, logistic sine chaos mapping was used to scramble the plain image. Afterwards, the scrambled image was divided into 2-by-2 sub blocks before encrypting each sub-block. Information entropy, correlation coefficient, key space analysis and the plaintext sensitivity of the technique revealed that the technique actually overcame the lack of diffusion limitation of existing chaotic image encryption algorithm. A novel way of hiding EMI inside medical images to be encrypted before being transmitted over a network was proposed by Masilamani [23]. Encryption/decryption and data hiding keys were embedded into a separate block of the secret message before a block-wise image encryption process was used to encrypt the cover image. A trained SVM model was used to extract the secret message along with the keys. The execution time and time complexity measured were on the higher side perhaps due to the complexity of the technique proposed. Ismail, Said, Radwan, Madian, and Abu-elyazeed [13] also proposed a generalized Double Humped (DH) logistic map using Pseudorandom Number Key (PNK) generation. The PNK was used in the encryption of biomedical images before sending them on a network. A very low correlation coefficients obtained from the original and encrypted images confirmed the encryption strength of the proposed technique. Also, the key sensitivity analysis of the encryption technique revealed that it is highly sensitive to every small change in the key parameters.

Furthermore, a two-dimensional (2D) logistic-sine-coupling map image encryption technique was proposed by Hua, Jin, Xu and Huang [11]. The technique designed a permutation algorithm to permutate the image pixels to different rows and columns after which a diffusion algorithm was used to spread the plain-image on the encrypted image. A higher encryption efficiency was recorded when the proposed technique was compared to other encryption techniques. Bai et al. [4] also proposed an encryption technique for securing EMI obtained from patients' Body Area Network (BAN). QRS complex of the electrocardiogram signal was used to extract vital signs from the patients' BAN, these vital signs were used to create the initial key before the key stream was generated using linear feedback shift register. Dynamic key updating and low energy consumption are the major advantages of the proposed technique. Blowfish algorithm has also been widely employed to secure medical images. such could be seen in the works of Kondawar and Gawali [17] who employed blowfish algorithm to secure EMI retrieved from a patient's wearable device. An ARM7 LPC2138 Microcontroller was programmed to synchronize the activities of several sensors connected to it.

However, blowfish algorithm was used to encrypt the retrieved data before forwarding it to the receiver section through the RF module. The secret key used for the encryption and decryption process would have been shared among the authorized medical personnel. Blowfish algorithm was also employed for textual information encryption by Raigoza [31]. The research aimed at determining if the size of original files do change after encryption. Also, they seek to know the point at which the execution time will change when ASCII values of input files change between 32 to 126. Results obtained revealed that the file size remains unchanged after encryption for input strings with lengths in multiples of 16 i.e. 16, 32, 48, 64 etc. Also, encryption and decryption times remains stable when there is a change in ASCII value. Furthermore, Panda [27] also employed blowfish to encrypt binary, text and image input files. Performance evaluation results obtained revealed that encryption time, decryption time and throughput increased with an increase in size when binary, text and image input files were used.

To improve the limitations of existing blowfish algorithm, several attempts at modifying blowfish algorithm have been reported in the literature. Nur, St, Darlis and Si [26] implemented a modified blowfish algorithm on Field-Programmable Gate Array (FPGA) for textual information encryption. The number of rounds required in the existing blowfish algorithm was reduced to 4 and 8 rounds while the key size was changed from 448 bit to 384 bit. Performance evaluation carried out revealed that lesser encryption time was recorded when 4 rounds were used compared to when 8 and 16 rounds were used. Also, lesser encryption rounds yielded greater throughput which means that a high encryption time will yield more throughputs. It was also observed that larger key lengths require more resources on FPGA.

Moreover, Hazra [10] employed blowfish and Diffie-Hellman techniques for image and file encryption. Blowfish algorithm was used to generate the secret key while using Diffie-Hellman protocol was used to generate a share private key to be used by the users over an unsecured network. With this technique, encrypted data can only be read by intended parties as a two level security system was developed. Similarly, a hybrid blowfish-MD5 and RSA-MD5 was presented by (Chauhan and Gupta [5]). Both hybridized algorithm was used for textual information encryption. Performance evaluation result obtained revealed that the size for the encrypted file increases for both hybridized algorithms but the execution time of Blowfish-MD5 was lesser when compared to RSA-MD5 algorithm. Blowfish algorithm was also modified for image encryption by Ali and Abead (2016). The multi sub-keys needed for encryption was generated from five instead of four S-boxes. The correlation coefficient of the algorithm and the number of pixels change rate revealed that the modified algorithm performed better than existing blowfish algorithm. Similarly, the F-Function of blowfish algorithm was optimized for encryption by Christina and Joe [8]. The F-function of the optimized blowfish algorithm has two S-boxes instead of four in the existing blowfish algorithm. Also, one XOR operation was used in the F-function in contrary to two additions and one XOR operations needed in the existing blowfish algorithm. However, a high execution time was recorded with the optimized blowfish technique though a high throughput was achieved. Conclusively, a comparative analysis of DES and blowfish algorithms was carried out by Nie and Zhang [25]. Results obtained revealed that the

existing blowfish algorithm encrypts faster than DES but it requires a larger memory than DES. This is certainly because of the memory required by the F-Function to generate the sub keys and authors were of the opinion that memory requirements could limit the use of blowfish for smart card applications and other memory constrained applications.

4 Methodology

To improve the existing blowfish algorithm, its F-function which is used to generate the round sub-keys was modified. Existing F-function in Eq. 1 uses two OR operations, one XOR operation and four s-boxes.

$$F(xL) = \left(\left(S_1 + S_2,\ mod\ 2^{32}\right) xor\ S_3\right) + S_4,\ mod\ 2^{32} \tag{1}$$

However, the modified F-function in Eq. 2 uses two XOR operations, one OR operation and four S-boxes.

$$F(xL) = \left(\left((S_1\ xor\ S_2) + S_3,\ mod 2^{32}\right) xor\ S_4\right) \tag{2}$$

The modified F-Function is further illustrate with Fig. 2.

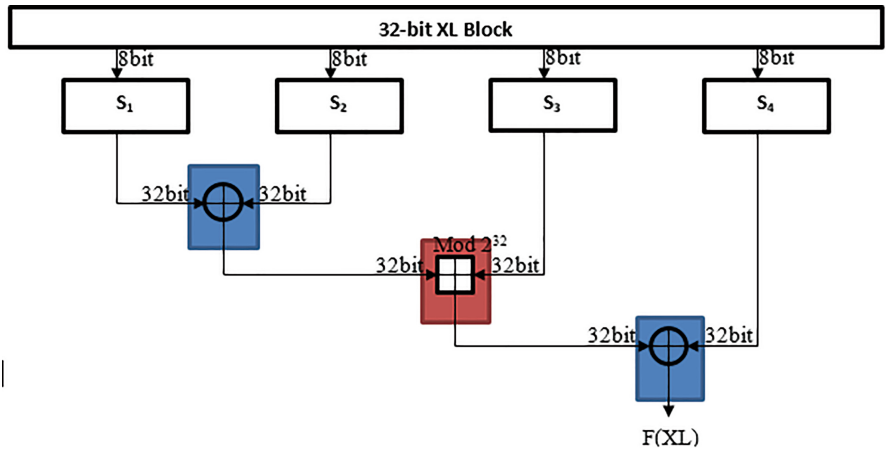

Fig. 2. F-Function of the modified blowfish algorithm

The pseudocode for the modified Blowfish Encryption is given as:

Step 1: Split the input 64bit data into xL and xR where each half contains 32-bit half data
Step 2: Repeat steps (a) to (c) while i = 1 to16

(a) Carry out XOR operation between xL and P[i].
(b) Compute the F-function of xL such that:

$$F(xL) = \left(((S_1 xor S_2) + S_3, mod 2^{32}) \, xor \, S_4\right)$$

XOR F(xL) with xR.

Step 3: Swap xR and xL.
Step 4: Carry out XOR operation between xR and P[16]
Step 5: Carry out XOR operation between xL and P[17].
Step 6: Merge xR and xL.

The pseudocode for the modified blowfish decryption is given as:

Step 1: Split the input 64bit data into xL and xR where each half contains 32-bit half data
Step 2: Repeat steps (a) to (c) while i = 16 to1

(a) Carry out XOR operation between xL and P[i].
(b) Compute the F-function of xL such that:

$$F(xL) = \left(((S_1 xor S_2) + S_3, mod 2^{32}) xor S_4\right)$$

F(xL) is XORed with xR.

Step 3: Swap xR and xL.
Step 4: Carry out XOR operation between xR and P[1]
Step 5: Carry out XOR operation between xL and P[0].
Step 6: Merge xR and xL.

The modified Blowfish Algorithm was employed to secure EMI of patients; textual and graphical information were considered. The implementation was carried out in MATLAB R2015a programming environment on a Dell Inspiron 15 N3000series; 500 GB, 4 GB RAM, core i3, 1.7 GHz Dual processor.

5 Results and Discussion

The graphical user interface used for EMI documentation is shown in Fig. 3. Patients medical records are captured using the interface. A patient id is automatically generated for each patient using the first letter of each name, the sex and their age. Also, an image can be uploaded to illustrate the patients' ailment. A secret key used for the encryption and decryption would have been distributed among authorized personnel. Once this key is supplied, the EMI can be encrypted as well as decrypted.

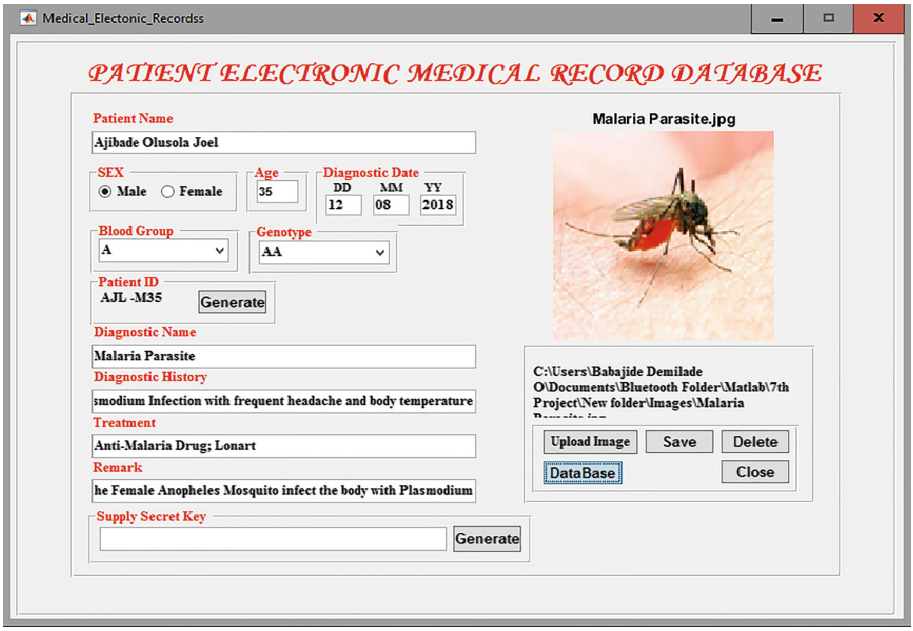

Fig. 3. Medical record interface

As expected, the saved EMI can also be retrieved by authorized medical personnel once the secret key has been supplied. Figure 4 illustrates an encrypted EMI that was decrypted.

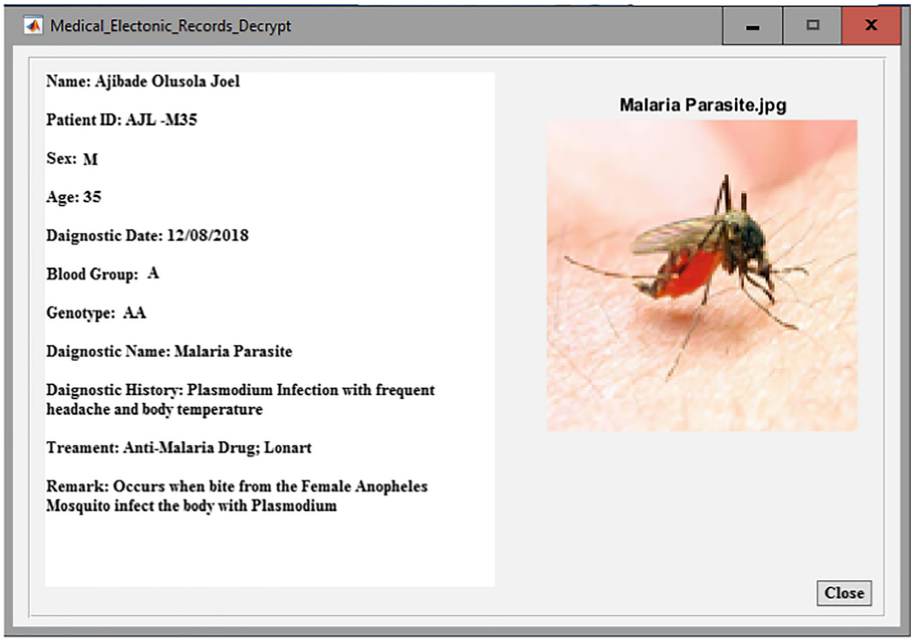

Fig. 4. Decrypted medical record

An attempt to access a decrypted EMI with a use a wrong secret key yields an unreadable message as shown in Fig. 5.

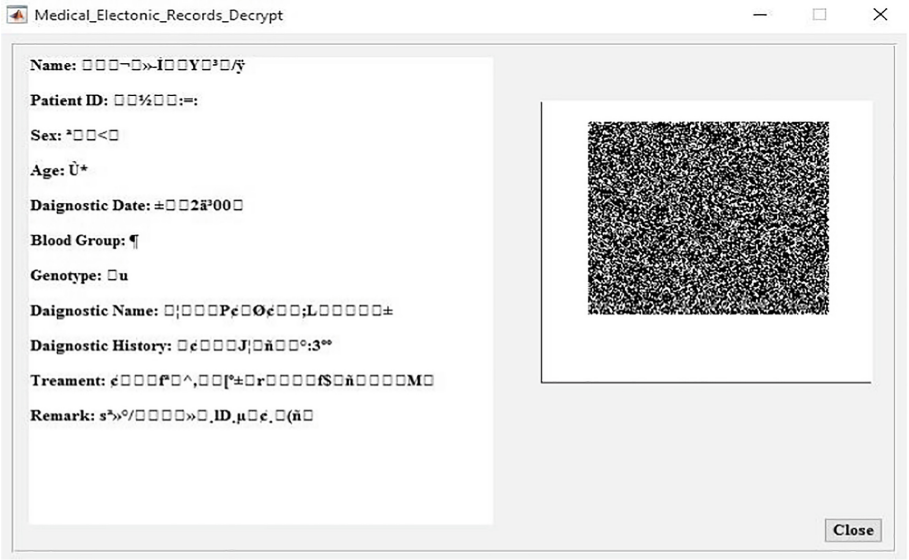

Fig. 5. Decrypted medical record with wrong secret key

6 Performance Evaluation

Depending on the amount of information available to the intruders, four classical types of attacks were reported by Ismail et al. [13]. Cipher text only attack occurs when an intruder attempts to deduce the secret key using the cipher text. Known plaintext attack occurs when the intruder attempts to deduce the secret key form the plain text and its corresponding cipher text available to him. Chosen plain text attack occurs when the intruder could access the encryption system, therefore an attempt is made to generate a cipher text with a plain text available to him. In the chosen cipher text attack, the intruder could access the decryption system, therefore an attempt is made to deduce a plain text from the cipher text available to him. Therefore, an encryption system must be evaluated against these attacks.

6.1 Differential Attack Analysis

This was used to confirm the strength of an encryption algorithm and also to ensure that the encryption system is resistive to any attacks that may arise from a compromise to the plain text or cipher text. Therefore, the diffusion performance of the algorithm could be determined using the Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). NPCR measures the change in pixel numbers between two images while UACI measures the average difference in intensity between two images. NPCR and UACI can be measured using Eqs. 3, 4 and 5 respectively:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100\% \tag{3}$$

$$D(i,j) = \begin{cases} 1, & c_1(i,j) \neq c_2(i,j) \\ 0 & c_1(i,j) = c_2(i,j) \end{cases} \tag{4}$$

$$UACI = \frac{1}{m \times n} \left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255} \right] \times 100\% \tag{5}$$

Where $P1(i,j)$ and $P2(i,j)$ are the $(i,j)th$ pixels of images $P1$ and $P2$ respectively, m and n are the pixels width.

The computed NPCR is 98.85% while UACI is 33.65% as against 99.6094% and 33.4635% proposed by Deng and Zhu [7]. This revealed that a slight change in the input image will noticeably alter the encrypted version. This confirms that the modified algorithm is key sensitive and resistive to differential attacks.

6.2 Execution Time of Modified Blowfish Against Existing Blowfish Algorithm

Execution time which measures the encryption and decryption time of the algorithm was used to compare the performance of modified blowfish algorithm against existing Blowfish algorithm. Comparative analysis of the execution time as shown in Fig. 6 revealed that the modified blowfish algorithm encrypts faster than existing blowfish algorithm. Also, the analysis of the decryption time shown in Fig. 7 revealed that the modified blowfish algorithm decrypts faster than existing blowfish algorithm.

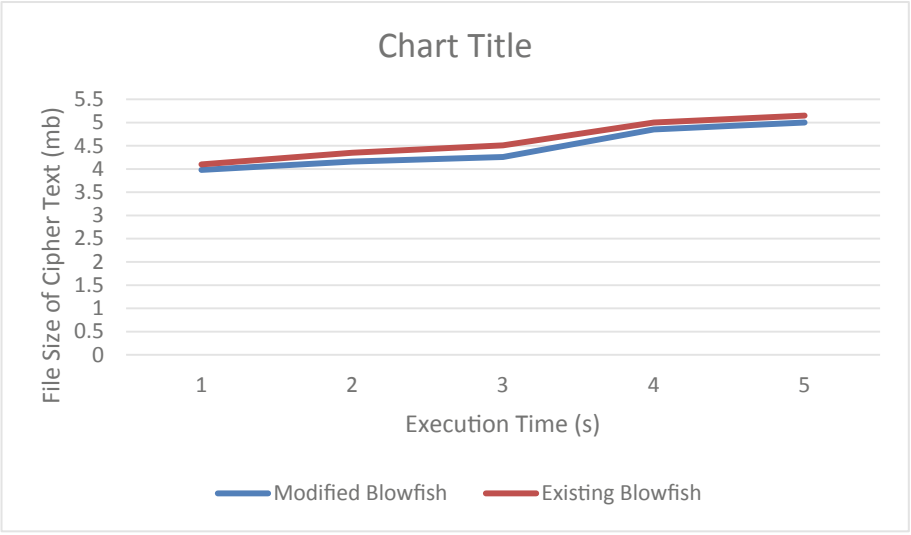

Fig. 6. Encryption time analysis

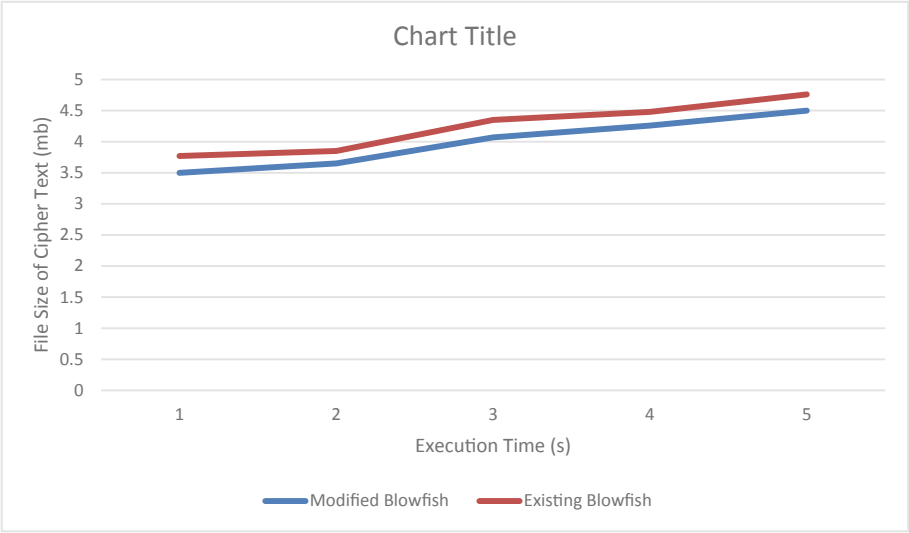

Fig. 7. Decryption time analysis

7 Conclusion

Electronic medical information is a sensitive information about the state of health of a patient and they need to be kept away from unauthorized individuals. Despite the limitations of blowfish algorithm, it can still keep these EMI secured from intruders. An attempt to improve the deficiencies of existing blowfish algorithm by modifying its F-function has been presented in this article. Differential Attack Analysis carried out has shown that the modified algorithm is secured against plaintext and cipher text attacks. Also, a comparative analysis between the modified and the existing blowfish algorithm has also shown that the modified version has a better execution time than the existing one. Other ways of improving the blowfish algorithm can be further explored towards increasing its encryption strength.

References

1. Christiana, A.O., Adeshola, G.Q., Oluwatobi, A.N.: Implementation of textual information encryption using 128, 192 and 256 bits advanced encryption standard algorithm. Ann. Comput. Sci. Ser. **15**(2), 153–159 (2017)
2. Alabaichi, A.M.: Security Analysis of Blowfish Algorithm, September 2013. https://doi.org/10.1109/ICoIA.2013.6650222
3. Andriole, K.P.: Security of electronic medical information and patient privacy: what you need to know. J. Am. Coll. Radiol. **11**(12), 1212–1216 (2014). https://doi.org/10.1016/j.jacr.2014.09.011
4. Bai, T., et al.: A lightweight method of data encryption in BANs using electrocardiogram signal. Future Gener. Comput. Syst. (2018). https://doi.org/10.1016/j.future.2018.01.031

5. Chauhan, A., Gupta, J.: A novel technique of cloud security based on hybrid encryption by Blowfish and MD5. In: 4th International Conference on Signal Processing, Computing and Control (ISPCC) (2017)
6. Chen, X., Hu, C.: Adaptive medical image encryption algorithm based on multiple chaotic mapping. Saudi J. Biol. Sci. **24**(8), 1821–1827 (2017). https://doi.org/10.1016/j.sjbs.2017.11.023
7. Deng, X.H., Zhu, C.X.: Image encryption algorithms based on chaos through dual scrambling of pixel position and bit. J. Commun. **35**(3), 216–223 (2014)
8. Christina, L., Joe, I.: Optimized Blowfish encryption technique. Int. J. Innov. Res. Comput. Commun. Eng. **2**(7), 5009–5015 (2014)
9. Gowda, S.N.: Using Blowfish encryption to enhance security feature of an image, vol. 200, pp. 126–129 (2016)
10. Hazra, T.K., Mahato, A., Mandal, A., Chakraborty, A.K.: A hybrid cryptosystem of image and text files using Blowfish and Diffie-Hellman techniques. In: 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), pp. 137–141 (2017)
11. Hua, Z., Jin, F., Xu, B., Huang, H.: 2D logistic-sine-coupling map for image encryption. Sig. Process. (2018). https://doi.org/10.1016/j.sigpro.2018.03.010
12. ICIT: Institute for Critical Infrastructure Technology. Hacking healthcare in 2016: lessons the healthcare industry can learn from the OPM breach (2016). http://icitech.org/wp-content/uploads/2016/01/ICIT-Brief-Hacking-Healthcare-IT-in-2016.pdf
13. Ismail, S.M., Said, L.A., Radwan, A.G., Madian, A.H., Abu-elyazeed, M.F.: Generalized double-humped logistic map-based medical image encryption. J. Adv. Res. **10**, 85–98 (2018). https://doi.org/10.1016/j.jare.2018.01.009
14. Jack, M.: Survey : 64 percent of patients use a digital device to manage health. Mobi Health News (2018)
15. Kaur, A., Singh, G.: A random selective block encryption technique for secure image cryptography using Blowfish algorithm. In: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1290–1293 (2018)
16. Keckley, P.H.: Privacy and security in health care : a fresh look. Deloitte Center for Health Solutions, pp. 1–20 (2013)
17. Kondawar, S.S., Gawali, D.H.: Blowfish algorithm for patient health monitoring. In: International Conference on Inventive Computation Technologies (ICICT), pp. 1–6 (2016)
18. KPMG: Health care and cyber security: increasing threats require increased capabilities (2015). https://assets.kpmg.com/content/dam/kpmg/pdf/2015/09/cyber-health-care-survey kpmg-2015.pdf
19. Landge, I.A.: VHDL based Blowfish implementation for secured embedded system design, pp. 3–7 (2017)
20. Langer, S.G.: Cyber-security issues in healthcare information technology. J. Dig. Imaging **2016**(October 2016), 117–125 (2017). https://doi.org/10.1007/s10278-016-9913-x
21. Lima, J.B., Madeiro, F., Sales, F.J.R.: Signal processing: image communication encryption of medical images based on the cosine number transform. Sig. Process. Image Commun. **35**, 1–8 (2015). https://doi.org/10.1016/j.image.2015.03.005
22. Martin, G., Martin, P., Hankin, C.: Cybersecurity and healthcare : how safe are we ? **3179**, 4–7 (2017). https://doi.org/10.1136/bmj.j3179
23. Masilamani, V.: ScienceDirect reversible reversible data data hiding hiding scheme scheme during during encryption encryption using using machine machine learning learning. Proc. Comput. Sci. **133**, 348–356 (2018). https://doi.org/10.1016/j.procs.2018.07.043
24. Netwrix Research Lab: 2017 IT Risks Report (2017)

25. Nie, T., Zhang, T.: A study of DES and Blowfish encryption algorithm. In: IEEE Region 10 Conference (TENCON 2009), pp. 1–4 (2009)
26. Nur, K., St, P., Darlis, D., Si, S.: An implementation of data encryption for Internet of Things using Blowfish algorithm on FPGA 2. In: 2nd International Conference on Information and Communication Technology (ICoICT), pp. 75–79 (2014)
27. Panda, M.: Performance Analysis of Encryption Algorithms for Security, pp. 278–284 (2016)
28. Park, E.H., Kim, J., Wile, L.L., Park, Y.S.: Factors affecting intention to disclose patientsÕ health information. Comput. Secur. (2018). https://doi.org/10.1016/j.cose.2018.05.003
29. Patel, P., Patel, R., Patel, N.: Integrated ECC and Blowfish for smartphone security. Proc. Comput. Sci. **78**(December 2015), 210–216 (2016). https://doi.org/10.1016/j.procs.2016.02.035
30. Quist-Aphetsi, K., Laurent, N., Anca, C.P., Sophie, G., Jojo, M.E., Nii, N.Q.: A cryptographic technique for security of medical images in health information systems. Proc. Comput. Sci. **58**, 538–543 (2015). https://doi.org/10.1016/j.procs.2015.08.070
31. Raigoza, J.: Evaluating performance of symmetric encryption algorithms, pp. 1378–1381 (2016). https://doi.org/10.1109/CSCI.2016.257
32. William, J.G., Adam, F., Adam, L.: Threats to information security—public health implications. New Engl. J. Med. **377**, 1–3 (2017)