International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

A Novel Encryption Algorithm by Fusion of Modified Blowfish Algorithm and Fermat's Little Theorem for Data Security

P.L.Chithra, K.Sathya

Abstract*: Nowadays, protecting sensitive factual information from hackers when it is transmitted over the broad network to communicate around the world is a huge task. To provide secure online transactions, communication, and business, researchers are insisted to develop novel security techniques. One of these approaches is on the basis of encrypting a plaintext will be in ciphertext by applying some mathematical models. This paper suggested a modified Blowfish algorithm with Fermat's Little Theorem to provide an additional layer of security for sensitive information while sending over a non-secure channel. The performance analysis of Modified Blowfish Algorithm and Fermat's Little Theorem (BFLT) encryption algorithm reveals consumption of memory, time consumption for encrypting and decrypting by comparing with the standard encryption algorithm. According to the analysis, this proposed encryption algorithm provides significantly upper level of performance and security and BFLT can be used for real-time implementations.

Key words: Blowfish, Fermat's Little Theorem, Encryption, Decryption, Fusion.

I. INTRODUCTION

Even though people are glad and comfortable with e-services, they meet many hazards now a day, by the crooked act like password hacking and personal information theft. Cryptography offers a numerous safety targets to secure the information while transfer through global network, non-modification of sensitive information and so forth. Result of the outstanding security benefits of cryptography is extensively applied everywhere today. The faster development of emerging new technology increases the worldwide communication network very simpler and also increases the prevalence and size of data transmission. While transmitting and receiving information through network protection of data is necessary against evil attacks. A standard encryption algorithm gives a higher level of protection to the public network. The Data Encryption Standard (DES)[4] uses 64 bits of data, every single 64 bits of data recurs from 1 to 16 times.

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author
P.L.Chithra*, Dept Computer Science, University of Madras, Chennai (Tamil Nadu) India. E-mail : chitrasp2001@yahoo.com
K.Sathya, Dept.of.Computer Science, University of Madras, Chennai (Tamil Nadu) India. E-mail : sathyabalaji33@gmail.com

It executes two functions with this input, bit shifting, and bit substitution furthermore this whole process is controlled by the key. In recent days DES and double-DES are considered as insecure but triple DES with three keys is suggested algorithm in NIST even though it takes more time than the AES algorithm. This algorithm (Advanced Encryption Standard) [3] possesses very simple mathematical function and still, it is widely in use. To offer more security, AES applies the following methods such as, mixing, key adding, substitution, transformation, permutation in each turn of AES but the last turn utilizes the four transformations [9].

Improved Proposed Encryption Standard (IPES) also called as IDEA is developed to replace the DES algorithm and was broken by the meet-in-the-middle attack as well as by narrow - bicliques attack. "Meet-in-the-middle is a type cryptanalytic attack that uses some sort of time-space trade-off to perform an attack like brute-force". "Biclique attack is a diverse of Meet-in-the-middle (MITM) type of cryptanalysis. It applies a biclique design to extend the number of possibly attacked laps by the MITM attacks". RC2 (Rivest's Cipher) is a block cipher cryptography algorithm which uses 64-bit block size with different key sizes and conversion of plaintext to ciphertext as well as ciphertext to plaintext done in 18 rounds. Block ciphers work on inflexible length string of bits which is called block. It is easily vulnerable to related-key attacks. "In cryptography, a related-key attack is any type of cryptanalysis where the attacker can perceive the operation of a cipher through several various keys whose scores are primarily unknown, but where few mathematical relationships merging the keys are well-known to the attacker".

Blowfish [5] is developed to replacing DES and IDEA which is used symmetric block cipher. It uses block size of 64-bit with diverse key sizes between 32 bit to 448 bit and it has 16 rounds. It works slowly in some of the applications and vulnerable to Birthday attacks in HTTPs and sweat32 attacks. Some asymmetric algorithms like Diffie-Hellman, RSA, and Elliptic Curve are available today for secure data transmission through network communication. RSA is the extensively used asymmetric algorithm currently, but Diffie - Hellman requires much CPU power and large data exchange for Digital signing and SSL. A hybrid encryption method that combines a novel encryption algorithm with a chaotic approach is the only solution to protect data in the future. This proposed system also combines the two novel (Modified Blowfish and Fermat's Little Theorem) technique which helps to increase the security of online communication.

II. LITERATURE REVIEW

The present situation utilizes cryptographic encryption algorithm which includes password encryption, encryption of mobile phone communication, encryption of smart card details, and DVDs. It has been infused day today life and also strongly used by major applications [1]. For safer communication via global network, information must be secured by the type of encryption. Encryption turns the information through the keys in the scrambled form.

The concern user possess access the key can decrypt the encrypted data [14]. The cryptographic algorithm can be categorized into two, an asymmetric (public) key and symmetric (private) key [8]. In symmetric key algorithms [6]: the same key is used to encrypting and decrypting the information by the transmitter and the receiver respectively. Krishna [17] implemented a novel mathematical method, in that method the output of the Elliptic Curve Cryptography (ECC) algorithm used different value and a dynamic time stamp to produce the cipher text. Triple-DES (block encryption algorithm) was enhanced and implemented from DES algorithm, applies 64-bit key comprising 56 efficient key bits and 8 parity bits. In triple-DES, the same DES encryption procedure is applied three times on the plaintext. In triple-DES the plaintext is converted to ciphertext with key 'A', converting ciphertext to plaintext with key 'B', and again plaintext is converted to ciphertext with key 'C'. Blowfish is a symmetric block cipher of 64-bit encryption algorithm with variable key length. This algorithm is functioning as two sections: key expansion section and data encryption section. The key expansion is used to transform a key of at most 448 bits into various arrays of sub keys totaling 4168 bytes [18].In the cryptography of asymmetric key, two separate keys are applied; secret key and public key although the two keys are connected mathematically. One key is applied to lock or convert the plaintext into ciphertext, and then the other key is applied to unlock or convert the ciphertext to plaintext. Modified ASCII Value (MAV) is created based on the ASCII values to strengthen the wedges encryption algorithm. Salts make cracking more difficult, because hacker's can't find the salt of the particular password [12]. Cryptographic algorithm is a basic mechanism for ensure the security of sensitive data. The main goal of applying encryption is used to prevent disclosure of information or maintain confidentiality in online communications. Encryption is like a person speaking to someone else while other people can listen, at the same time other people cannot understand what person is conversing [13].Public key perhaps made public without adversely affecting protection, while the private key should not be disclosed to anyone and not approved to read the messages [11]. Singh and Gilhotra [2] developed a cryptographic algorithm that converts the plaintext data into ciphertext data in four phases. By using this algorithm, a given text file is converted as a floating-point within 0 and 1. RSA cryptography is widely accepted public-key algorithm [7]. RSA is the first, and most widely applied asymmetric algorithm. RSA is named because of the three mathematicians who implemented it, Rivest, Shamir, and Adleman. RSA is applied to secure the software products and it can be utilized for secure key exchange, safeguard the digital signatures and helps to encrypt the small packets of data. DES is 64-bit size block cipher, with a 56-bit key. 16-round series of substitution and permutation is used in DES. In each and every round, the data and key bits are shifted, permutated, XORed, and sent through 8 s-boxes, as well as a set of lookup tables that are mandatory to the DES algorithm. Converting ciphertext to plaintext is exactly the same process, which are performed in reverse of encryption process [16].

RC4 is also the type of symmetric key with stream cipher encryption created by Ron Rivest and it is familiar for its simpleness as well as for its swiftness. It utilizes a variable key length within 40 to 2048 bits. In this algorithm, the data stream is XORed on the created key sequence [10]. A study was performed among different existing popular secret key algorithms such as DES, AES, and Blowfish. These algorithms were implemented, and evaluated their performance by comparing the input files of varying contents and sizes [15].

III. PROPOSED ENCRYPTION ALGORITHM

This encryption algorithm is a fusion of the modified Blowfish algorithm and Fermat's Little Theorem (BFLT). It consists of 64-bit size of block and key length varies from 32 to 448 bits with 16 round Feistel cipher. In addition to that, this cipher is applied to Fermat's little theorem to provide an extra layer of security. Decrypting the given file is the reverse process of encrypting the same file. The original key array size is 576 bits long and the XOR operations among the keys. It gives protection against brute force attack because of the BFLT's key changing behavior. Key changing is the XOR operation between the original key with the initial sub-key (K1), again this XOR operation continues with the original key to producing 15 sub-keys. In this paper, JavaScript is used to carry out the implementation. For this experiment single machine is used with the following specifications: RAM 4 GB and x64 processor with a speed of 2.67 GHz. The first phase of this proposed algorithm is the creation of 15 sub keys (K2, K3,….K16) from the original key. These keys are identical to encryption and decryption in the server and client sides of the network. The cryptographic strength depends on the number of keys which are used for encrypting the text and decrypting. More the number of keys produce more security to the encryption technique. The entire block cipher encryption algorithms in the literature are using 1 key. In BFLT, one original key and one master sub key (2 keys) are used to strengthen the algorithm. In Table 2, we analysed the number of keys used by the existing block cipher algorithms. In this proposed algorithm, the second phase consists of the following. The modified Blowfish is used to divide the plain text into two subparts (PL and PR), the left part of the plain text (PL) is XOR with the key K1 and right part also XOR with the same key (K1).

Both the PLXOR (32 bit) and PRXOR (32 bit) combined together as a single ciphertext (CT1, 64 bit). The third phase of this proposed scheme is used by Fermat's Little Theorem. This theorem receives the combined text (CT1, 64 bit) as input, then the ciphertext CT2 is calculated by finding the power CT1 with the prime number 'p'.

This CT2 is modulus with the prime number to get the 'r'. According to the 'r', either CT3 is received after divided CT2 with 'p' otherwise CT1 is taken as CT3.

This same process is continued until the K16 is used. Then this CT3 is decrypted in the same way likewise the process of encryption. BFLT encryption and decryption algorithm are given below as Algorithm1 and 2.

Algorithm 1: BFLT Encryption Algorithm Round1

function *Encryption ()*
item PL;
for $I \rightarrow 0$ *to* PL_{len} ***do***
$PL_{len1} \rightarrow Hex\ (PL_{len})$
end for
$mid \leftarrow Math.floor(PL_{len1}$ ***divide*** *2)*
$L \leftarrow PL_{len1}\ (0, mid)$
$R \leftarrow PL_{len1}\ (mid)$
$O \leftarrow O_k$
for $I \rightarrow 0$ *to* O_k*length* ***do***
$PL_{len2} \leftarrow Hex\ (O_k)$
end for
$K_1 \leftarrow subkey1$
for $I \rightarrow 0$ *to* K_1*length* ***do***
$PL_{len3} \leftarrow Hex\ (K_1)$
end for
$p1_{xor} \leftarrow (PL_{len2} \oplus PL_{len3})$
$L_{xor} \leftarrow (L \oplus p1_{xor})$
$R_{xor} \leftarrow (R \oplus p1_{xor})$
$CT_1 \leftarrow L_{xor} + ' ' + R_{xor}$
$P \leftarrow prime$
$CT_2 \leftarrow$ ***power*** (CT_1, p)
$r \leftarrow (CT_2$ ***mod*** $p)$
if $r == CT_1$ ***then***
$CT_3 \leftarrow CT_2$ ***divide*** p
else
$CT_3 \leftarrow CT_1$
End if
Return CT_3

Algorithm 2: BFLT Decryption Algorithm Round1

Function *Decryption ()*
$P \leftarrow prime$
Item CT_3
$CT_2 \leftarrow$ ***power*** (CT_3, p)
$R \leftarrow (CT_2$ ***mod*** $p)$
If $r == CT_3$ ***then***
$CT_1 \leftarrow CT_3$ ***multiply*** p
else
$CT_1 \leftarrow CT_3$
end if
$mid \leftarrow Math.floor\ (CT_1length$ ***divide*** $2)$
$L_{xor} \leftarrow CT_1\ (0,\ mid)$
$R_{xor} \leftarrow CT_1\ (mid)$
item $PL_{len3},\ PL_{len2}$
for $I \rightarrow 0$ *to* PL_{len2}*length* ***do***
$O_k \leftarrow Hex\ (PL_{len2})$
End for
for $I \rightarrow 0$ *to* K_1*length*
$PL_{len3} \leftarrow Hex\ (K_1)$
End for
$p1_{xor} \leftarrow (O_k \oplus K_1)$
$R \leftarrow (R_{xor} \oplus p1_{xor})$
$L \leftarrow (L_{xor} \oplus p1_{xor})$
$PL_{len1} \leftarrow L + ' ' + R$
for $I \rightarrow 0$ PL_{len1} *length* ***do***
$PL \leftarrow char\ (PL_{len1})$
return *PL*
End function

IV. EXPERIMENTAL RESULT AND PERFORMANCE ANALYSIS OF BFLT ALGORITHM

Communicating with each other with no time is the outstanding significance in our current era. Countless research studies are performed on computer networks to promote the communication to the next stage. At the same time secure communication is one of the main research fields which is increasingly important every day. Many researches have been performed so far to sustain communication security. The performance of BFLT is evaluated by implementing the algorithm in JAVASCRIPT using Net Beans IDE 8.0. Encryption algorithm performs an important role in security of online communication where encryption time, memory usage for output and battery power. The selected encryption algorithm such as AES, DES, RC6 and RC5 are used for performance evaluation.

Based on the text files used for encryption by BFLT, the experimental result was concluded that BFLT algorithm consumes least encryption time than DES, and BFLT consumes least memory for encrypting the data. Encryption time difference is very minor in case of AES algorithm and DES algorithm while compare with BFLT. RSA consumes very long encryption time and high memory consumption but output byte is least in case of RSA algorithm. To measure the efficiency of the proposed algorithm, we analyzed the memory usage and time consumption for encryption and decryption by comparing it with other standard encryption algorithms.

A.Memory consumption analysis

Consumption of memory for program and data are very essential requirement for actual implementations with confined hardware resources such as microcontrollers. If the memory consumption is less for data and program, the rate of usage in actual implementations is high and the fewer the cost. In real-time implementations, memory consumption for data and program are essential parts. If memory usage decreases, the usage in real environment implementations also increases. Table.1 shows the consumption of memory for program and data by BFLT compared with the existing cryptography algorithms. According to the Table.1 DES algorithm consumes 2986 bytes memory for program and 762 for data which is huge to compare with existing algorithms. On the other hand, BFLT occupies 1455 bytes memory for program and 13 bytes for data.

B. Time consumption analysis

Time consumption for encryption and decryption is another important criterion in cryptography techniques. Besides less memory utilization and possible use in real environment implementations, another substantial criterion is speed. Speed always depends to a great extent on the exact model of the processor and on the software. BFLT consumes only 43 microseconds for both encryption as well as decryption which is highly close to the AES cryptography algorithm. The efficient encryption algorithm takes extremely low encryption and decryption time.

Cryptographic strength depends on the Key length. A large key secures the cipher to a wider extent but encryption speed is inversely proportional to the key length.

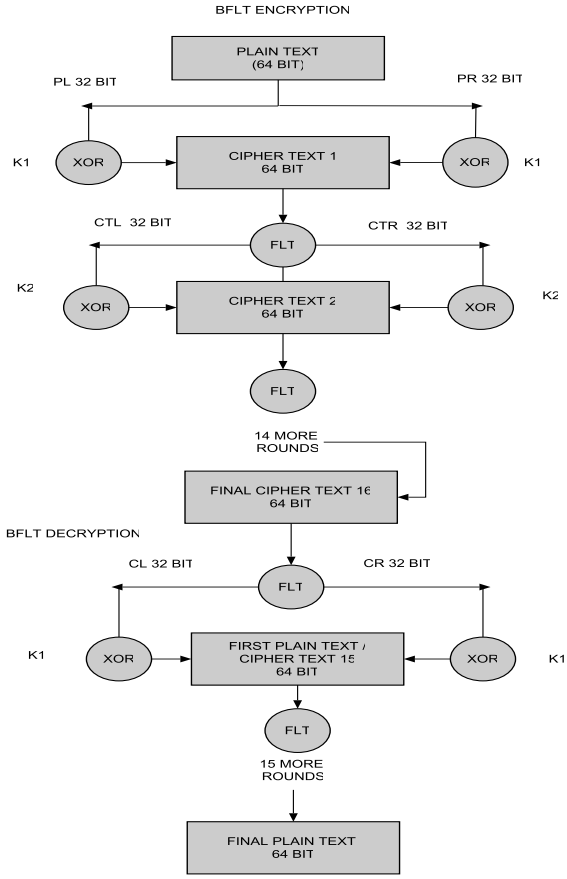

Fig.1. Flow diagram of BFLT Encryption and Decryption algorithm.

C.Key size analysis

If the number of bits increases in the key size, the encryption speed will be slow. While designing BFLT, we consider both speed and security according to the key length. Hence, BFLT is created with a 64-bit key size both for encryption and decryption.

D.Throughput analysis

Performance of the encryption algorithm also depends on the throughput of the algorithm. If more will be the throughput of the encryption algorithm, higher will be its performance. The throughput is calculated with the help of the formula 1.

$$Throughput = \frac{Size\ of\ Plain\ Text}{Encryption\ Time} \quad \ldots \quad (1)$$

Throughput is calculated from the analysis done for encryption time which is calculated as 43 microseconds (0.043 milliseconds) with 1 KB plaintext. It can be observed in Table 3. Here we have compared memory usage of program and data of BFLT with other algorithm like AES, RC6, RC5 and DES. Data memory and program memory of the BFLT is considerably less than other existing bench mark algorithms.

Table.1. Memory usage of program and data

Algorithm	Program memory (bytes)	Data memory (bytes)
BFLT	1455	13
AES	1486	18
RC6	2706	23
RC5	2710	22
DES	2986	762

Table.2. Number of keys used in the Algorithm

S.No	Algorithm	No. of. Keys
1.	BFLT	2
2.	DES	1
3.	RC2	1
4.	RC5	1
5.	BLOWFISH	1

Table.3. Throughput of encryption produced by BFLT and other symmetric algorithms

S.No	Algorithm	Throughput(kb/ms)
1.	BFLT	23.26
2.	DES	3.882
3.	RC2	3.321
4.	RC6	7.343
5.	BLOWFISH	13.307

V.CONCLUSION

Internet is particularly used by Individuals, Co-operatives and Governments to transfer information for communicate each other. While transfer data through global network chances are open to hack the information. In order to secure information, we must encrypt or decrypt information with the help of cryptography algorithms. A fusion of encryption algorithm (BFLT) is proposed in this paper. This algorithm is based on the Blowfish encryption algorithm and Fermat's Little Theorem. Enrichments are applied on the Blowfish algorithm by adding Fermat's Little Theorem mathematical model instead of using Feistel Function F in the Blowfish algorithm. This tiny enrichment eliminates the slowness of the Blowfish algorithm significantly, and the performance analysis is done with parameters like program memory, data memory, encryption time and throughput. According to the performance analysis, BFLT performs well than the existing commonly used algorithms.

REFERENCES

1. Shashi Mehrotra Seth, Rajan Mishra, "Comparative Analysis of Encryption Algorithms for Data Communication", Int Journal of Comp Sci and Technology, Vol. 2, (2), 2011.p.292-294.
2. Singh, A. and Gilhotra, R. Data security using private key encryption system based on arithmetic coding. Int Journal of Net Sec and its App (IJNSA)., 3, 2011, 58-67. https://pdfs.semanticscholar.org/96c0/be3d7374c426723cc62d43b63ac6624db413.pdf.
3. AES Algorithm. (Available from: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf) [Accessed on 6 July 2015].
4. DES Algorithm. (Available from: http://csrc.nist.gov/ publications/fips/fips46-3/fips46-3.pdf) [Accessed on 6 July 2015].
5. Schneier B. Description of a new variable-length key, 64-bit block cipher (Blowfish). Fast Software Encryption, Cambridge Security Workshop Proceedings, Springer-Verlag: Cambridge, UK, 1993; 191–204.
6. N. Kumar and S. Agrawal, "An efficient and effective lossless symmetric key cryptography algorithm for an image," 2014 IEEE Int Conf on Adv in Eng & Tech Research (ICAETR - 2014), Unnao, 2014, pp.1-5. doi:10.1109/ICAETR.2014.7012788
7. R. Rivest, A. Shamir, L. Adleman. "A method for obtaining digital signatures and public-key crypto systems" z. Communications of the ACM, Feb 1978.

8. Diaasalama Abd Elminaam, Hatem Mohamad Abdual Kader, Mohly Mohamed Hadhoud, "Evaluation the Performance of Symmetric Encryption Algorithms", Int journal of net sec. vol.10,No.3, pp,216-222, May 2010.
9. Neetu Settia. "Cryptanalysis of modern Cryptography Algorithms". Int Journal of Comp Sci and Tech. December 2010.
10. Rashmi N, Jyothi K," An Improved Method for Reversible Data Hiding Steganography Combined with Cryptography", Proceedings of the II Int Conf on Inventive Sys and Contl. (ICISC 2018) IEEE Xplore, p.81-84.
11. Preetha, M. Nithya." A Study and Performance Analysis of RSA Algorithm", Int Journal of Comp Sci and Mob Computing, Vol. 2(6), 2013, p.126-139.
12. PL.Chithra, K.Sathya, "A Novel Password Encryption Using Wedges Algorithm with QR Code", Int Journal of Pure and Applied Maths Volume 119 No. 7 2018, 857-861.
13. Marshall D.Abrams, Harold J.podell on Cryptography.
14. Anoop MS, "Public key Cryptography (Applications Algorithm and Mathematical Explanations)".
15. A.Nadeem, "A performance comparison of data encryption algorithms", IEEE info and comm technologies, pp.84-89, 2006.
16. Erik Olson, Woojin Yu, "Encryption for Mobile Computing".
17. Krishna, A.V. Time stamp based ECC encryption and decryption. Int Arab Journal of Info Technology. 11, 2014, 276281. http://ccis2k.org/iajit/PDF/vol.11,no.3/4571.pdf
18. "Blowfish Algorithm" Available: http://www.schneier.com/blowfish. html

AUTHORS PROFILE

Dr. PL. Chithra is the Professor in the Department of Computer Science at the University of Madras. She received her M.C.A and Ph.D. degrees from Alagappa University, Tamil Nadu, India and University of Madras, Tamil Nadu, India respectively. She has more than 29 years of experience in teaching. She has been serving as Organizing Chair and Program Chair of several International conferences, Program Committees of several International conferences and she is awarded UGC FIP program for two years. Ph.D. and M.Phil. research supervisor for Guiding Image Processing Techniques, Big data analytics and Network Security. She has conducted several refresher courses and published more than 67 papers in national and international journals with 42 citations and 4 h-index. She is one of the Computer science staff selection committee members for various affiliated colleges of University of Madras.

Mrs. K. Sathya is a Research Scholar in the Department of Computer Science at the University of Madras. She received M.C.A, M.Phil degree from Madurai Kamaraj University and Thiruvalluvar University, Tamil Nadu, India respectively. She has qualified National Eligibility Test (NET). She is specialized in Information Security and her main research area includes Cryptography and Network Security, Information Security through encryption, creation of passwords and development of novel CAPTCHAs.