ISSN No:-2456-2165

Modified Blowfish Algorithm

Adwaith Samod T
Department of Computer Science and Engineering
Mar Athanasius College of Engineering
Ernakulam, India

Devika TV
Department of Computer Science and Engineering
Mar Athanasius College of Engineering
Ernakulam, India

Hareesh P
Department of Computer Science and Engineering
Mar Athanasius College of Engineering
Ernakulam, India

Leya Elizabeth Sunny
Assistant Professor, Department of Computer Science and Engineering
Mar Athanasius College of Engineering
Ernakulam, India

Abstract:- **Every time a sender and receiver communicate, se-curity has always been a major concern. Numerous cryptographic techniques, such as AES (Advanced Encryption Standard), Data Encryption Standard (DES), Triple DES (3DES), Blowfish, etc., are employed to prevent security breaches. We make an effort to change the current Blowfish method based on various factors, including time, the avalanche effect, and non linearity.**

By combining the XOR and addition operations employed in the original approach, the ”f” function is adjusted. Sbox is also changed. Text, image, audio, and video files are encrypted and decrypted using the modified blowfish method, which is then tested. We generate and examine nine cases. The findings of all the tests performed on these cases point to the conclusion that the modified Blowfish method is more compact and safe than the earlier because of the security of the modified algorithm with various cases.

Keywords:- *Modified Blowfish Algorithm, Encryption, F-function, Multimedia Encryption.*

I. INTRODUCTION

Information security has emerged as a crucial concern in business, industry, and administration as a result of the rapid growth in digital communication and the sharing of electronic data. Security is the main concern for every communication between sender and recipient in the modern era. Sender and recipient would suffer significant losses if there are any security lapses during communication. Today’s cryptography provides a variety of crucial methods for securing data and safeguarding information.

By applying specific mathematical computational proce-dures to change text into unintelligible form, cryptography offers a way to protect sensitive information. With the right key, the text may then be transformed back into readable form. Utilizing cryptography provides information communication that is secure, private, and confidential. Data privacy is guar-anteed by cryptographic methods using symmetric and asym-metric encryption. Popularly used cryptographic algorithms for symmetric encryption include DES, 3DES, AES, Rivest Shamir Adleman (RSA), and Blowfish. Each has strengths and weaknesses. Experimental findings and comparisons showed that the Blowfish algorithm was the best among these when taking time into account.

The Blowfish algorithm was initially developed by Bruce Schneier in 1994 to replace the antiquated DES. The 64-bit symmetric key block cypher with changeable length is a distinguishing feature of blowfish. With the exception of when changing keys, Blowfish provides a free alternative to currently used encryption algorithms that have varying levels of security. To test the security feature and speed provided by Blowfish, numerous studies conducted performance comparisons based on various assessment factors. The findings revealed that it is definitely quick and secure.

The current standard requires a minimum of 128-bit block size, which makes Blowfish unsuitable because it can only accommodate 64-bit blocks. This property is seen as undesirable because it may lead to duplicate blocks that will eventually make other types of attacks possible, compromising data security. Blowfish is regarded as a remarkably fast block cypher. Although Twofish, a Blowfish-related technique, supports 128-bit block sizes and offers a high level of security, it is slower than Blowfish at encrypting data. The block size of Blowfish has been extended to 128 bits by a number of researchers; however, the results show a significant increase in time and a requirement for more memory, which makes the performance less favourable for use in applications that prioritise speed and renders it inefficient for use in small devices with limited memory.

II. RELATED WORKS

A. Derivation Cases

Two s-boxes instead of one prevents symmetry in the pro-cess. The process retained the overall structure of the blowfish algorithm but provided two derivation in the f function. The second derivation proved to have better performance in terms of avalanche effect and time. The time for key generation is

independent of the input size while the time for encryption and decryption depends upon the file size. This algorithm with 128 bit block size and 128 bit key size can be used to encrypt text, image and other types in addition to electronic medical data.

B. Text Cryptography based on Modified Blowfish Algorithm

Text in digital platforms can be encrypted using modified blowfish algorithm and Lempel-Ziv-Welch (LZW). Blowfish uses variable keylength of 32 to 448 bits. Modified blowfish is the fastest out of the two in encryption and decryption. These algorithms provide scope for combining both these algorithms to provide better security. These algorithms provide fast encryption and decryption without compromising security in digital platforms such as e-commerce portals, online marketing, electronic banking etc.

C. Message encryption technique based on enhanced blowfish algorithm

By lowering the number of rounds and raising the block length with a constant length during encryption and decryption, as well as adding a transformation method on some rounds, an improved version of the Blowfish algorithm is created. The Blowfish Algorithm has been improved, resulting in a significant reduction in execution time without sacrificing the complexity of the encrypted file content. The algorithm runs more quickly because of the reduction in iterations. Additionally, the complexity of the encrypted file's content was increased by block size improvements and other manipulations such the use of byte splitting, block size transposition, shift rows, and mix rows.

III. BACKGROUND

A. Cryptography

The study of secure communication methods, such as encryption, that only the message's sender and intended recipient can access, is known as cryptography. The word is derived from kryptos, a concealed word in Greek. It is closely related to encryption, which is the process of converting plain text into ciphertext before sending it and then back again after receiving it. The obscuring of information in photographs using methods like microdots or merging is also covered by cryptography. The most typical use of cryptography is to encrypt and decrypt email and other plain-text messages while sending electronic data.

B. Blowfish Algorithm

Blowfish algorithm is a symmetric block cipher. A secret key with a variable length that can be any length up to 448 bits is used with a 64-bit data block cypher. The algorithm consists of pieces for both data encryption and key expansion. The expansion key converts a key with a maximum length of 448 bits into several subkey arrays with a combined length of 4168 bytes. Data is encrypted using a 16-round Feistel Network. It has a key-dependent permutation and a key-and-data-dependent switch for each round. For all operations, a 32-bit word addition and XOR is used. The only new operations for the four indexed arrays are data lookups each round.

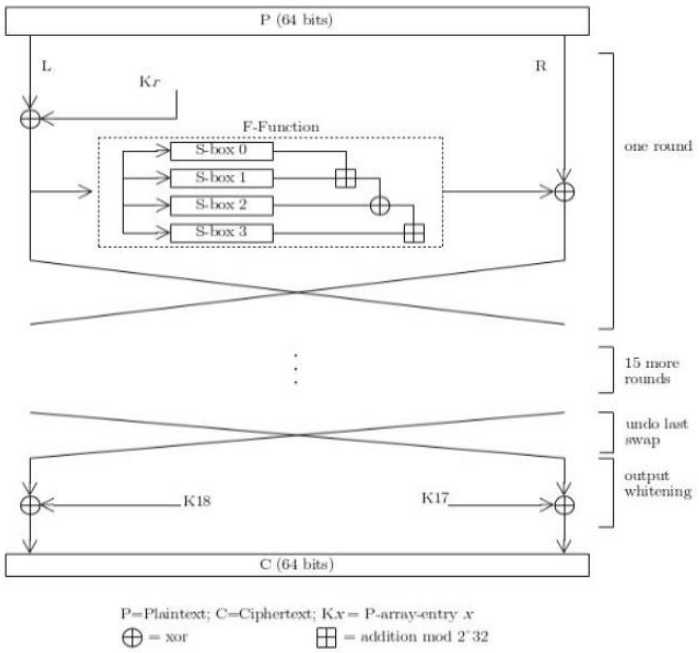

Fig. 1. Fiestel Structure of Blowfish

C. Multimedia encryption

In order to create an interactive presentation, multimedia mixes several content types as text, audio, photos, animations, and video. Security of multimedia applications is a crucial and difficult topic since they are especially easy to intercept over wireless networks. The primary enabling technology for maintaining secrecy and preventing unauthorised access to the content is multimedia encryption.

D. Fisher-Yates Shuffle

Ronald Fisher and Frank Yates proposed the Fisher-Yates shuffle (FYS), a technique for producing a random permutation of a finite linear array. Every variation of FYS in an array produces results that are identical. The Fisher-Yates shuffle in its contemporary form is introduced by Richard Durstenfeld. Donald E. Knuth, in his groundbreaking book The Art of Computer Programming, popularised the work of Durstenfeld. The more efficient version of today is an in-place shuffle, which uses no additional storage space and only time proportionate to the number of elements shuffled.

—To shuffle an array a of n elements (indices 0...n-1):

for i from n-1 down to 1 do

j ← random integer such that 0 j i exchange a[j] and a[i]

ISSN No:-2456-2165

E. Avalanche effect in Cryptography

The term ”avalanche effect” refers to a particular manner in which mathematical operations employed in encryption behave. One of the desirable properties of any encryption scheme is the avalanche effect. A small change in the plain text or the key should cause a large change in the encrypted text. Avalanche effect is the name for this characteristic. It measures the impact of a modest alteration to the plain text or the key on the cipher-text. The following relation should always be met by a decent encryption algorithm:

Avalanche effect > 50%

The effect makes sure that a plain-text cannot be readily predicted by an attacker using statistical analysis. That is, the encrypted text can be easily decrypted if a change in a single bit of the input causes a change in only a single bit of the desired output.

F. Non Linearity

Boolean function non-linearity standards are divided into groups based on how well they work with cryptographic architecture. A criterion invariant is left by the largest transformation group while classifying an object. These functions concurrently have minimum correlation to affine functions, maximum distance to linear structures, and both maximum distance to affine functions and maximum distance to structures. Frequently, the Walsh-Hadamard Transform (WHT), which makes use of the sign function, is used to assess the nonlinearity. The highest level of nonlinearity is 120.

IV. IMPLEMENTATION

A. Blowfish Algorithm

64 bit input plaintext is divided into 32 bit left(LE0) and right (RE0). LE0 is XORed to P1 in the P-array. All entries in the P-array are 32-bit. The result is given as input to the F-function. The output of the F-function is XOR ed with right(RE0). The current left and right are interchanged. This process continues until the number of rounds reaches 16 or its corresponding value. Finally the left and right are XOR ed with the last two values of the P-array and XORed with each other to produce the cipher text. The reverse process is used to decrypt the cipher text into the plaintext. Round: A round consists of mainly four functions

- LE0 is XORed with the number of P-array (corresponding to the number of round)
- The output is given as input to the F-function
- The output from the F-function is XORed with RE0 gives next RE
- LE and RE s are exchanged

(In the funcion the S-box consists of 256 entries of 32 bit each)

B. S-box permutation

The existing S-box of blowfish algorithm is shuffled using Fisher Yates Shuffle algorithm for generating a random per-muation of the S-box. This randomness can enhance the non linearity parameter of the S-box.

C. Case 1

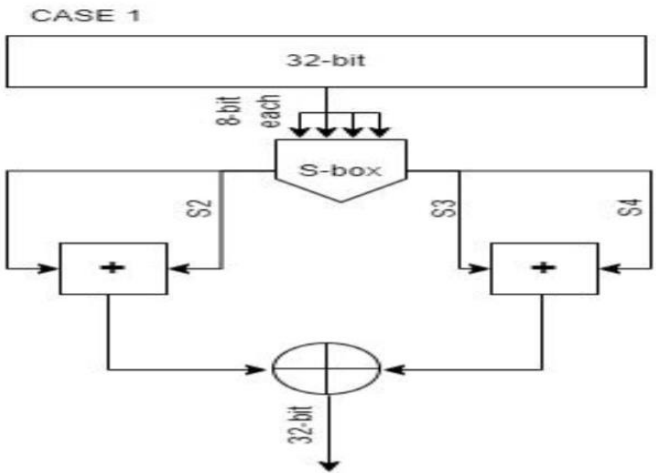

Fig. 2. Case 1

In this case F(xL) can be calculated as:

$F(xL)=((S1 + S2) \bmod 2^{32}) XOR ((S3+S4) \bmod 2^{32})$

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is ADD ed with second and XOR ed with the result of ADD function between the third and fourth elements. This gives a 32 bit text which is the output of the function.

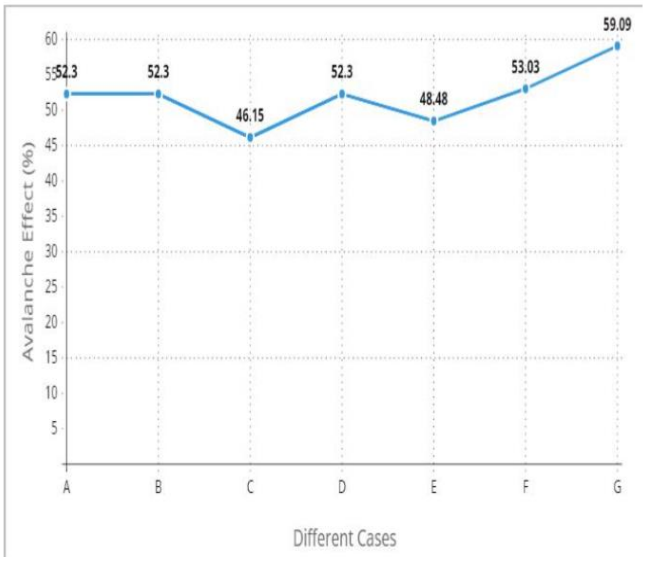

Fig. 3. Case 1 Avalanche Effects

ISSN No:-2456-2165

D. Case 2

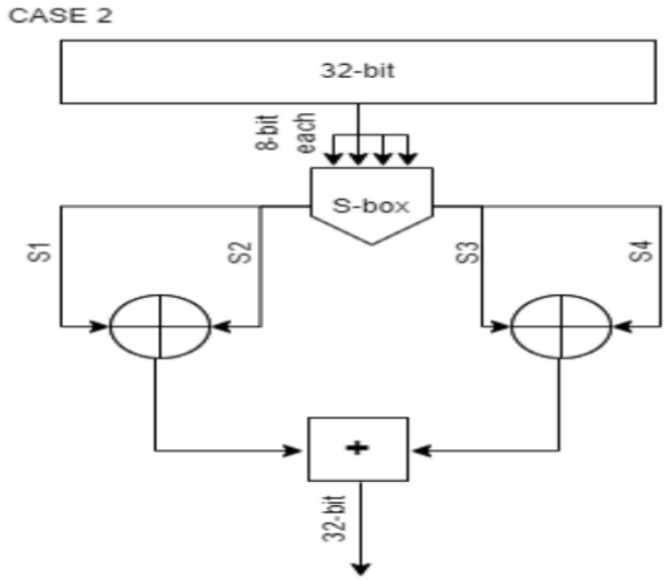

Fig. 4. Case 2

In this case F(xL) can be calculated as:

$F(xL)=((S1 \text{ XOR } S2) \bmod 2^{32}2)+((S3 \text{ XOR } S4) \bmod 2^{32})$

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is XOR ed with second and ADD ed with the result of XOR function between the third and fourth elements. This gives a 32 bit text which is the output of the function.

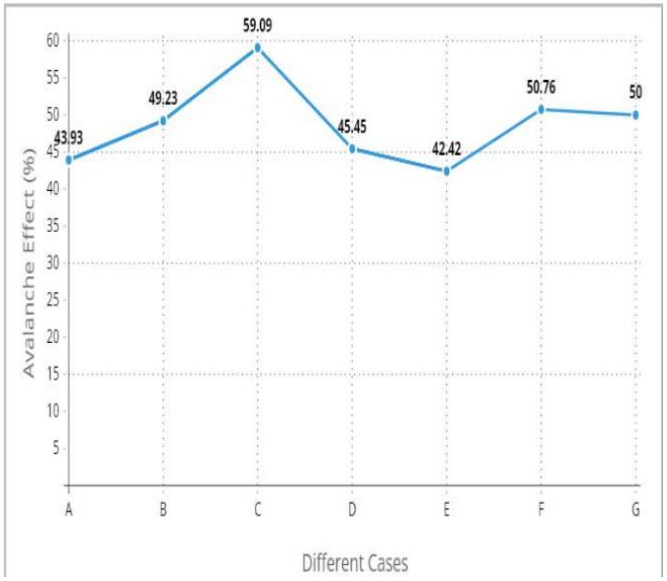

Fig. 5. Case 2 Avalanche Effects

E. Case 3

In this case F(xL) can be calculated as:

$F(xL)=(((S1 + S3) \bmod 2^{32}))XOR((S2+S4) \bmod 2^{32})$

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is ADD ed with third and XOR ed with the result of ADD function between the second and fourth elements. This gives a 32 bit text which is the output of the function.

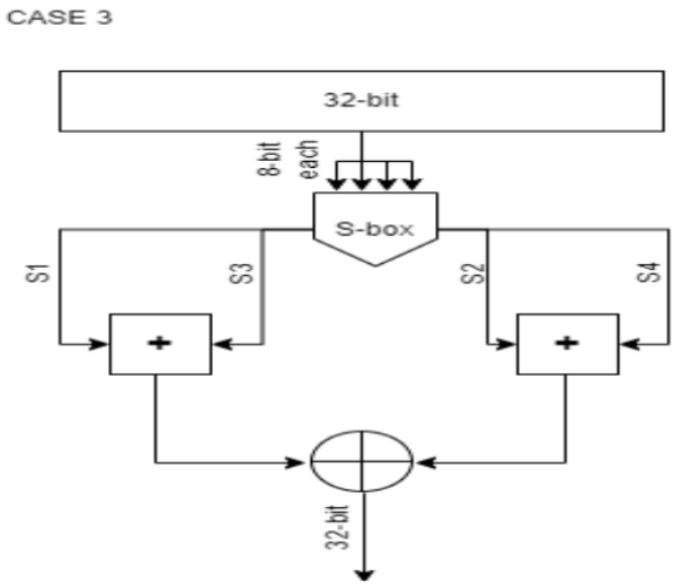

Fig. 6. Case 3

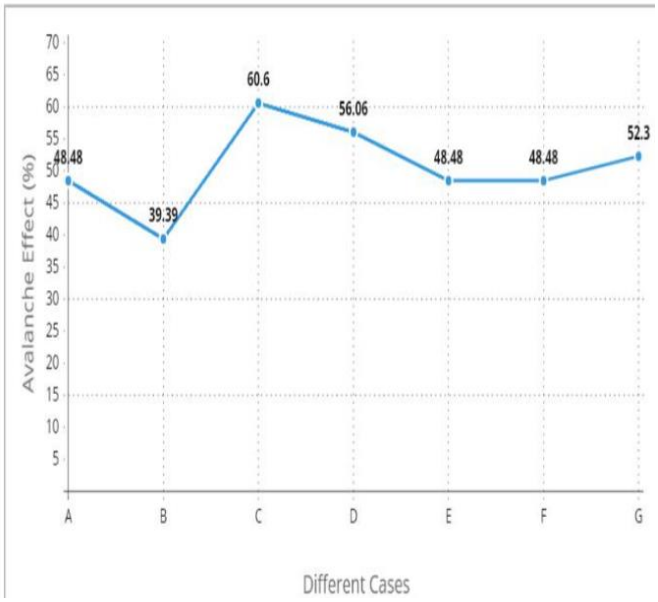

Fig. 7. Case 3 Avalanche Effects

F. Case 4

In this case F(xL) can be calculated as:

$F(xL)=(((S1 \text{ XOR } S3) \bmod 2^{32}))+((S2 \text{ XOR } S4) \bmod 2^{32})$.

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is XOR ed with third and ADD ed with the result of XOR function between the second and fourth elements. This gives a 32 bit text which is the output of the function.

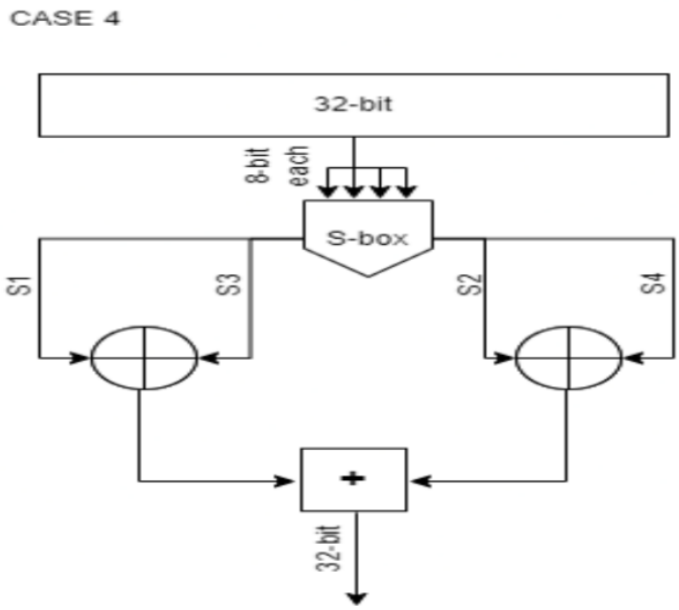

Fig. 8. Case 4

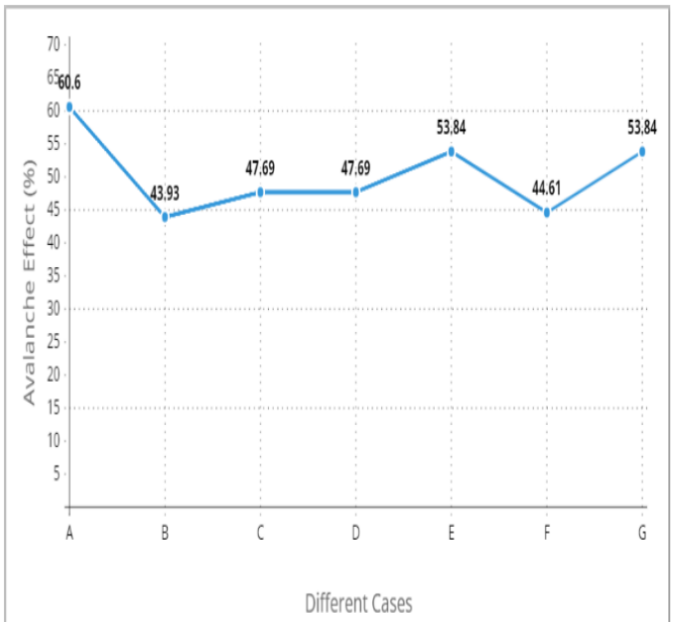

Fig. 9. Case 4 Avalanche Effects

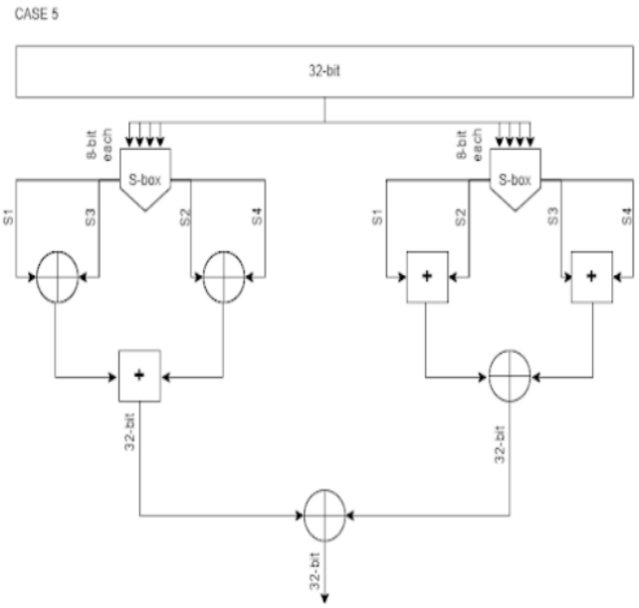

Fig. 10. Case 5

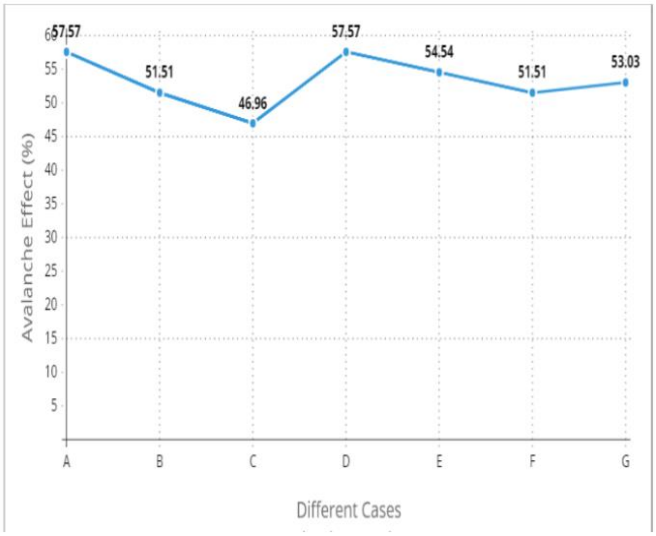

Fig. 11. Case 5 Avalanche Effects

G. Case 5

In this case F(xL) can be calculated as:

$F(xL)=(((S1 + S2) \bmod 2^{32}))XOR((S3+S4) \bmod 2^{32})) \hat{} (((S1 \text{ XOR } S3) \bmod 2^{32}))+((S2 \text{ XOR } S4) \bmod 2^{32}))$

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is ADD ed with second and XOR ed with the result of ADD function between the third and fourth elements.The result is then XORed with the result of the first element from S-box is XOR ed with third and ADD ed with the result of XOR function between the second and fourth elements. This gives a 32 bit text which is the output of the function.

ISSN No:-2456-2165

H. Case 6

In this case F(xL) can be calculated as:

$F(xL)=(((S1 + S2)\bmod 2^{32}))XOR((S3+S4)\bmod 2^{32})) \hat{} (((S1 XOR S2)\bmod 2^{32})+((S3 XOR S4)\bmod 2^{32}))$

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is ADD ed with second and XOR ed with the result of ADD function between the third and fourth elements. The result is then XORed with the result of the first element from S-box is XOR ed with second and ADD ed with the result of XOR function between the third and fourth elements. This gives a 32 bit text which is the output of the function.

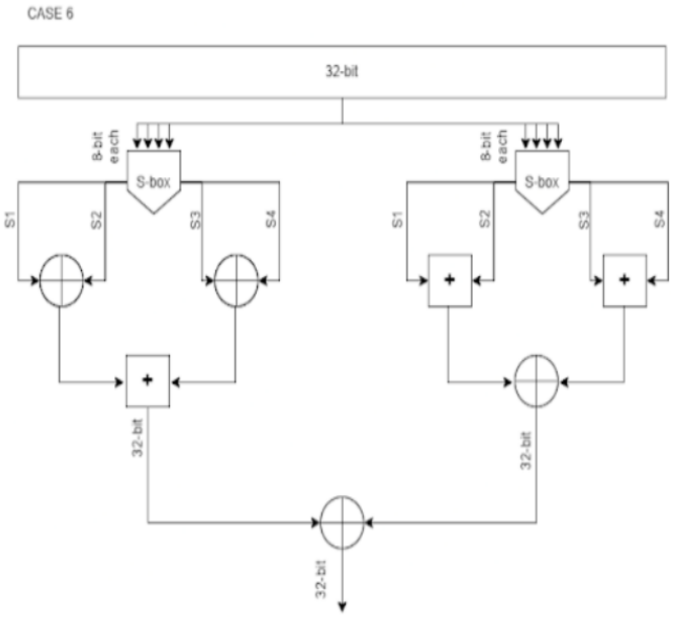

Fig. 12. Case 6

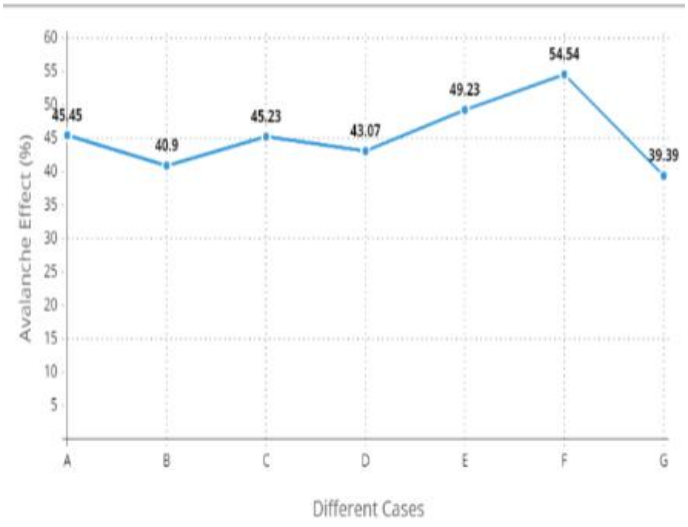

Fig. 13. Case 6 Avalanche Effects

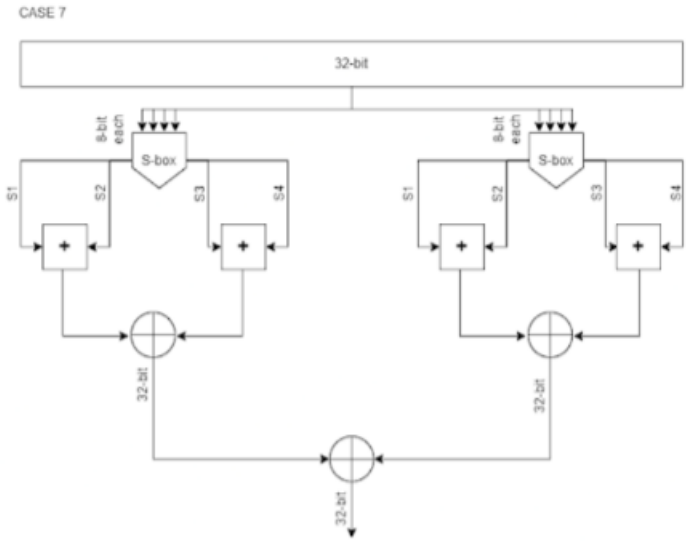

Fig. 14. Case 7

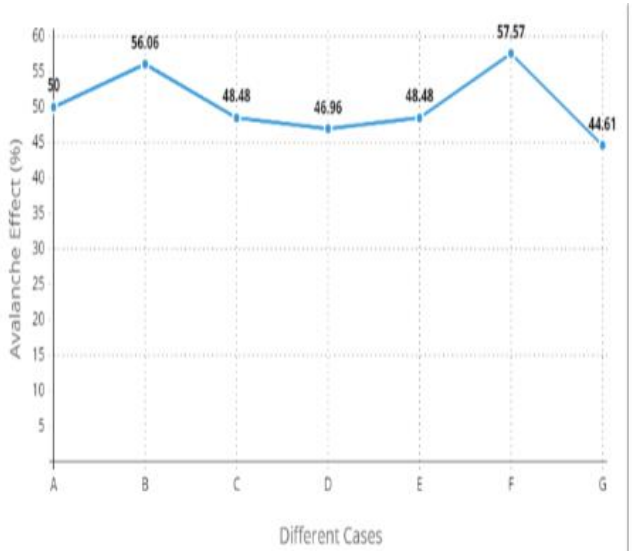

Fig. 15. Case 7 Avalanche Effects

I. Case 7

In this case F(xL) can be calculated as:

$F(xL)=(((S1 + S2)\bmod 2^{3}2))XOR((S3+S4)\bmod 2^{3}2)) \hat{} (((S1 + S2)\bmod 2^{3}2))XOR((S3+S4)\bmod 2^{3}2))$

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is ADD ed with second and XOR ed with the result of ADD function between the third and fourth elements. The result is then XORed with the result of the first element from S-box is ADD ed with second and XOR ed with the result of ADD function between the third and fourth elements. This gives a 32 bit text which is the output of the function.

J. Case 8

In this case F(xL) can be calculated as:

$F(xL)=(((S1 + S2) \bmod 2^{32}))XOR((S3+S4) \bmod 2^{32})) \hat{} (((S1 + S3) \bmod 2^{32}))XOR((S2+S4) \bmod 2^{32}))$

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is ADD ed with second and XOR ed with the result of ADD function between the third and fourth elements. the result is then XORed with the result of the first element from S-box is ADDed with third and XOR ed with the result of ADD function between the second and fourth elements. This gives a 32 bit text which is the output of the function.

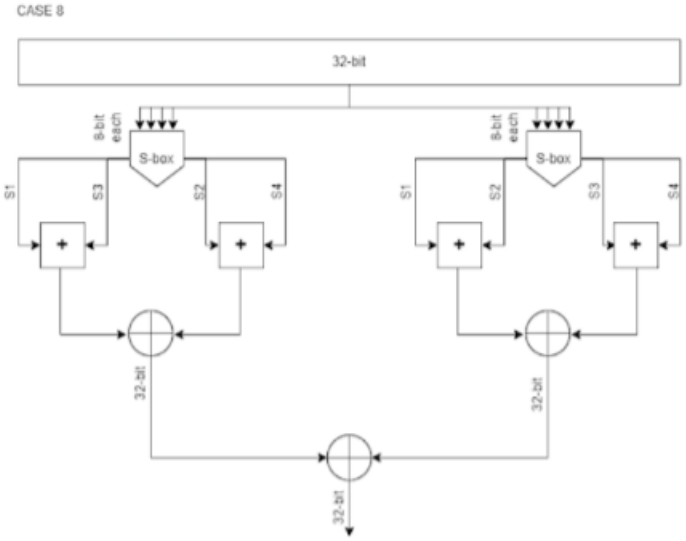

Fig. 16. Case 8

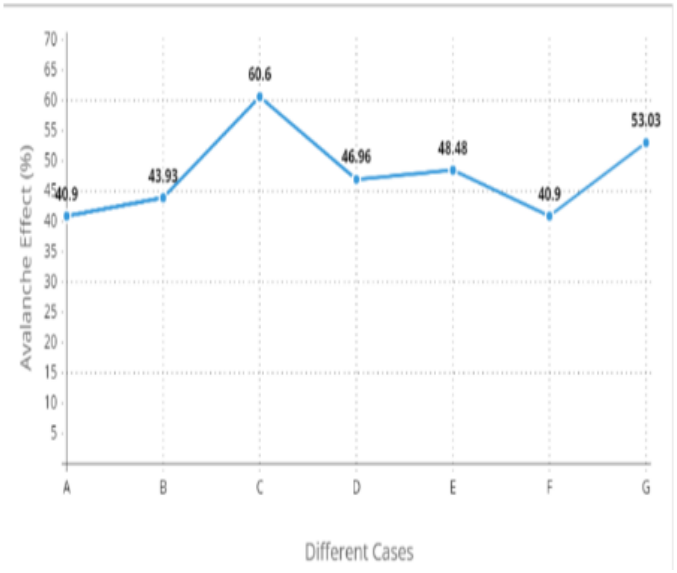

Fig. 17. Case 8 Avalanche Effects

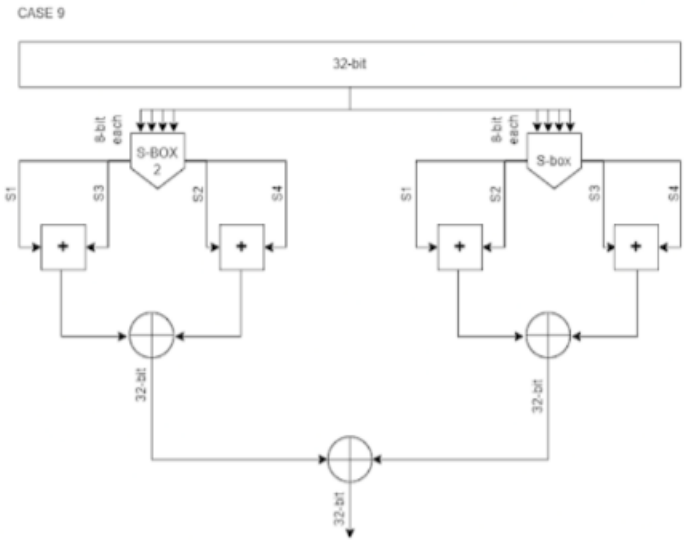

Fig. 18. Case 9

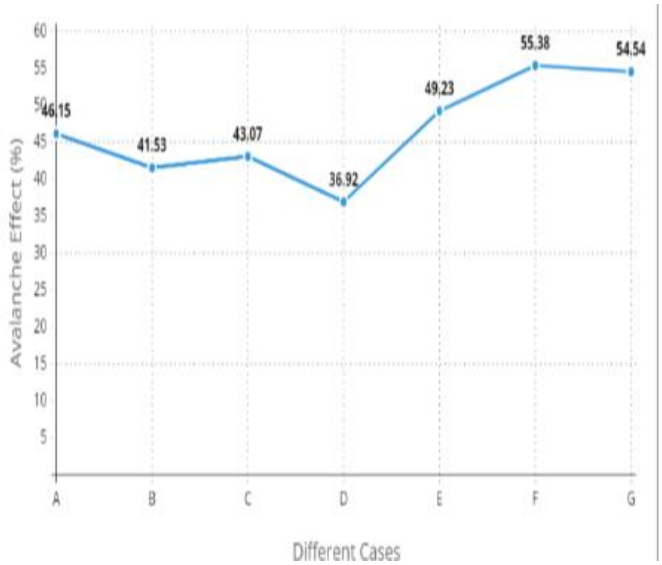

Fig. 19. Case 9 Avalanche Effects

K. Case 9

In this case F(xL) can be calculated as:

$F(xL)=(((S1 + S2) \bmod 2^{32}))XOR((S3+S4) \bmod 2^{32})) \hat{} (((S11 + S13) \bmod 2^{32}))XOR((S12+S14) \bmod 2^{32}))$

The 32 bit text is split into 4 parts with 8 bit each and used to find corresponding S-box 32 bit values for each. Then the first element from S-box is ADD ed with second and XOR ed with the result of ADD function between the third and fourth elements. the result is then XORed with the result of the first element from S-box2 is XOR ed with third and ADD ed with the result of XOR function between the second and fourth elements. This gives a 32 bit text which is the output of the function.

ISSN No:-2456-2165

V. RESULT

The proposed algorithm uses a different f function and s-box to encrypt and decrypt different types of files like text, images, audio and video. The avalanche effect and non-linearity is found to be slightly increased than original blowfish algorithm since the proposed algorithm includes complex calculations which increases the security. The modified blowfish algorithm obtained avalanche effect of 53.246 percentage and non lin-earity of 117.343.

VI. FUTURE RESEARCH

Promising results of modified blowfish algorithm shows the possibility of testing its application in cloud computing. Testing out the implication of the modified algorithm by exploiting a real world application would show the significance of the performance.

Fig. 20. Avalanche effects comparison

PARAMETERS	BLOWFISH ALGORITHM	MODIFIED BLOWFISH ALGORITHM
TIME FOR ENCRYPTION AND DECRYPTION(s)	0.53	0.55
AVG AVALANCHE EFFECT	51.954	53.246
NON LINEARITY	116.312	117.343

Fig. 21. Analysis of Modified Algorithm

VII. CONCLUSION

Modification have been done in the f function of the blow-fish algorithm to test out its implication in the parameters of the algorithm for comparison. The use of modified f function and s-box to encrypt and decrypt different types of files like text, images, audio and video has shown promising results in the performance of Blowfish Algorithm by increasing avalanche effect and non-linearity with little to no difference in time. Different cases of f-function have been implemented to illustrate and compare the different modification of the blowfish algorithm and the results suggest that case 5 with an average avalanche effect of 53.24 tend to perform better than the others.

REFERENCES

[1]. V. Poonia and N. S. Yadav, ”Analysis of modified Blowfish algorithm in different cases with various parameters,” 2015 International Conference on Advanced Computing and Communication Systems, 2015, pp. 1-5
[2]. Corpuz, Reynaldo Gerardo, Bobby Medina, Ruji. (2018). A Modified Approach of Blowfish Algorithm Based On S-Box Permutation using Shuffle Algorithm. 140-145. 10.1145/3301326.3301331.
[3]. Quilala, Theda Flare Quilala, Rogel. (2021). Modified Blowfish algorithm analysis using derivation cases. Bulletin of Electrical Engineering and Informatics. 10. 2192-2200. 10.11591/eei.v10i4.2292.
[4]. S. Akhter and M. B. Chowdhury, ”Bangla and English Text Cryptography Based on Modified Blowfish and Lempel-Ziv-Welch Algorithm to Minimize Execution Time,” 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), 2019, pp. 96-101
[5]. M. Ali, Nada Abead, Suaad. (2016). Modified Blowfish Algorithm for Image Encryption using Multi Keys based on five Sboxes. 57. 2968-2978.
[6]. Godfrey L Dulla (2019). A unique message encryption technique based on enhanced blowfish algorithm 2019 IOP Conf. Ser.: Mater. Sci. Eng. 482 012001
[7]. B. K. Maram and J. M. Gnanasekar, —Evaluation of Key Dependent S-Box Based Data Security Algorithm using Hamming Distance and Balanced Output,Journal, vol.5, no. 1, pp. 67–75, 2016
[8]. U. C¸avus¸oglu,˘ A. Zengin, I. Pehlivan, and S. Kac¸ar, — A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system,|| Nonlinear Dyn., vol. 87, no. 2, pp. 1081–1094, 2017
[9]. M. Eberl, —The Fisher – Yates shuffle,|| pp. 1–9, 2018.
[10]. M. Ahmad, P. M. Khan, and M. Z. Ansari, —A Simple and Efficient Key-Dependent S-Box Design Using Fisher-Yates Shuffle Technique,|| Commun. Comput. Inf. Sci., vol. 420 CCIS, pp. 540–550, 2014