A Novel Feistel-based Cryptosystem for Text Encryption

Akhil Kaushik
Assistant Professor, CE Department
T.I.T&S College
Bhiwani, Haryana, India

Satvika
Assistant Professor, CE Department
T.I.T&S College
Bhiwani, Haryana, India

ABSTRACT
Ever since the internet was employed for commercial applications, it has become unsafe place for exchanging data. The mission critical data of any organization needs a security mechanism which can guard against malicious attacks. Most of these security problems are intentionally caused by notorious people trying to gain some financial benefit or harm someone. Every organization utilizes some kind of safety measures to avoid access to unauthorized users like firewalls, proxy servers, access lists, antiviruses etc. Other important application layer security mechanisms that are widely used are cryptography and steganography. Steganography can be described as the technique for hiding a secret message within a larger one such that no one except the sender and receiver suspects the contents or even the existence of the hidden message. On the contrary, cryptography deals with altering the message contents such that they become meaningless if anyone intercepts them while transmission. In this paper, a novel text cryptographic algorithm named FBC Cryptosystem have been proposed for meeting the requirements of secure data transfer. The proposed algorithm is also compared to the de-facto encryption standards to foresee its performance and security aspects. It is found out that the proposed algorithm is a secure and proficient way for data encryption.

Keywords
Cryptography, Decryption, Encryption Key, Feistel Based Cryptosystem, Symmetric key algorithm.

1. INTRODUCTION
The internet began with the research and academic people that used it for sending emails and sharing resources. But with advent of time, the commercial aspect of internet took over the globe like a hurricane. This huge change introduced the idea of cryptography in the corporate environment. Earlier cryptography was only used by military, lovers, diarists and diplomatic corps. Military played the biggest role in development of cryptography, which then evolved to provide security against illicit attacks in computer and communication systems [1]. Traditionally, the ciphers used to be character-oriented, but modern ciphers have to encrypt not only text, but the whole information (which may include graphics, numbers, audio, video data, etc.). The modern ciphers also exhibit another difference from the traditional ciphers, as they are based on a special feature known as the "Feistel Function". Most of modern ciphers like DES, 3DES, etc., are based on feistel function; and have been accepted as industry standard for non-military applications. The popular encryption standards like DES, AES, IDEA, etc are block encryption algorithms i.e. they take a block of plaintext for encryption and produce a block of ciphertext. Encoding block of data rather than stream of data makes the encryption process faster and thus more useful for commercial applications [2]. However, they may suffer from replay attacks i.e. produce the same ciphertext for the same plaintext, if the encryption algorithm is in Electronic Code Book(ECB) mode. Large keys and message spaces are necessary but not sufficient for the block cipher to be secure [3]. It is important that the implemented transformation be complex, and be a mapping into a large set (the ciphertext-space) without an easily recognized structure.

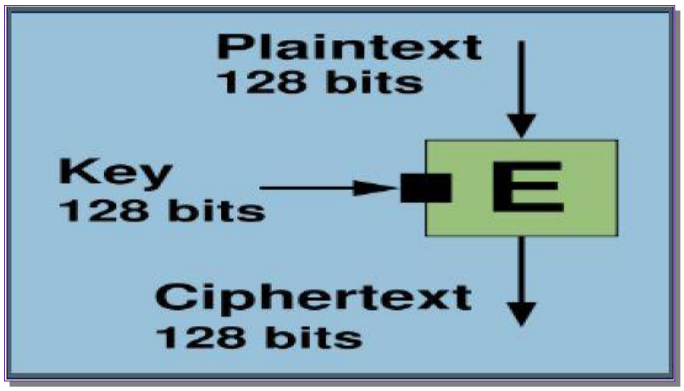

Fig 1: Functioning of Block Encryption

Most of the popular cryptographic algorithms fall under Cipher Block Chaining (CBC) mode. This mode eliminates some problems of ECB mode by including the previous cipher block in encrypting the current block. Even though the key remains same for cryptosystem, equal blocks of plaintext produce different size blocks of ciphertext [4]. Hence, the complexity of algorithm increases and makes it hard to crack. One issue that needs to be addressed here is that of error control. A transmission error in one block will introduce errors in following blocks as they are dependent on each other. An encryption/ decryption error will also set up whole bunch of errors in the whole encryption/ decryption process, finally proving catastrophic. Thus, choosing the proper mode of operation is very important while designing and developing any cryptographic algorithm [5]. The next section describes the proposed algorithm - Feistel Based Cryptosystem (FBC) for enhanced security in detail.

2. PROPOSED SYMMETRIC KEY ALGORITHM
Feistel Based Cryptosystem (FBC) uses symmetric encryption approach i.e. using same key for encoding & decoding. FBC is based on Feistel structure. The idea of Feistel structure is basically taken from Data Encryption standard (DES) and since then it has become a basis for the development of modern ciphers. The Feistel structure ensures that decryption and encryption are very similar processes; the only difference is that the sub-keys are applied in the reverse order when decrypting. The feistel function is normally defined as the combination of multiple binary operations that may include Substitution-boxes, Permutation-boxes, etc.

FBC is based on the thought of using intermediate ciphertext of previous block for encryption of the next block of data i.e. Cipher Block Chaining (CBC) mode. It ensures that the blocks will be interdependent and hence the

complexity is increased. But it also means that an extra caution needs to be taken while transmission and encryption to avoid chain of errors.

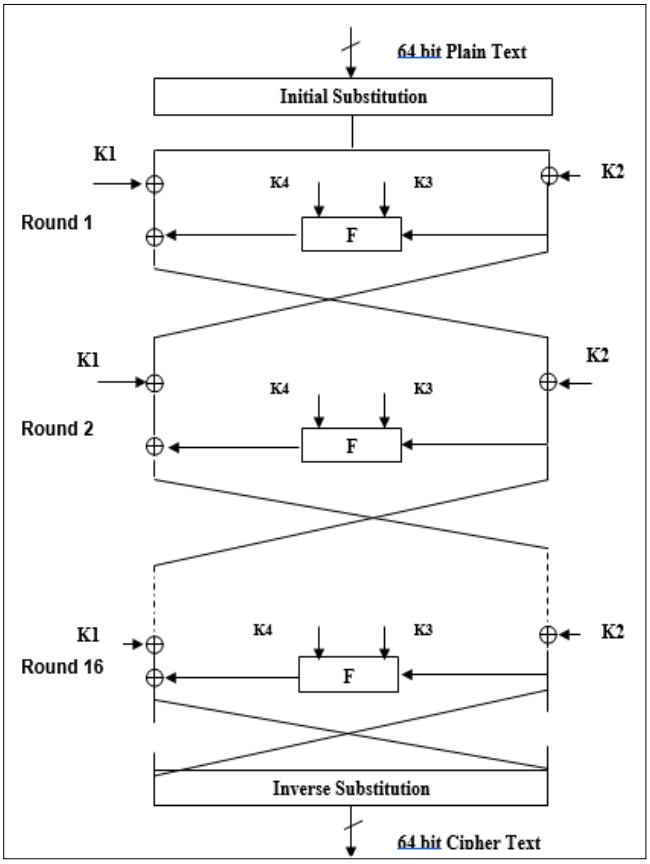

Fig 2: Working of Feistel Based Cipher

Most of the renowned block-cipher algorithms adhere to the Feistel cipher structure, with the exception of AES. The newly suggested algorithm also follows this structure. It functions as a block cipher, operating on 64-bit plaintext blocks with a 128-bit key. This algorithm serves for both encryption and decryption purposes. The 64-bit data block splits into two 32-bit sub-blocks: XL and XR, serving as input for the initial round. The algorithm comprises a total of 16 rounds, each utilizing four 32-bit subkeys. The algorithm employs 64 subkeys (4 for each of the 16 rounds). Initially, the 128-bit key is divided into four 32-bit subkeys, which serve as the four subkeys for the first round. Subsequently, the key is shifted 8 bits to the left, divided into four subkeys again, and this process continues until the end of the algorithm.

The processing of plaintext advances through four phases. Initially, the plain text undergoes an initial substitution in its two 32-bit sub blocks. Subsequently, this process is succeeded by a phase comprising 16 rounds of the identical function. Then, the left and right halves of the resulting output are interchanged. Ultimately, the output undergoes an inverse substitution to generate the 64-bit Cipher text.

Table I. Substitution Table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	61	62	63	64	65	66	186	187	188
1	190	191	192	193	194	50	51	52	53
2	55	56	57	176	177	178	179	180	181
3	183	184	185	40	41	42	43	44	45
4	47	48	49	166	167	168	169	170	171
5	173	174	175	31	32	33	34	35	36
6	152	153	154	155	156	157	16	17	18
7	20	21	22	141	142	143	144	145	146
8	148	149	8	9	10	11	12	13	14

2.1 Initial Substitution

The substitution table is derived from the ASCII table, initially segmented into 33 blocks (0 – 32) based on character types. The creation of the substitution table (Table 1) involves employing the permutation value and following the procedure outlined previously. Within this table, the row denotes the left digit, and the column signifies the right digit. For instance, when the plaintext is "K," with an ASCII value of 75, it transforms into 143 (row 7 and column 5) utilizing the aforementioned substitution table. Similarly, all plaintext is converted using the initial substitution method.

2.2 Round Function

The round function is basically divided into 2 sub-functions: Folding and XOR using Even-Odd Function, which are described as follows: Before the folding function, the 32-bit data is first XORed with sub-key (K3). Then, the folding function is applied to the intermediate output which writes the plaintext in a two-dimensional matrix and rewrites it using vertical folding and then horizontal folding. Subsequently, the data is then divided into even and odd bits, which are then XORed using respective even and odd bits of K4 subkey, which are finally remerged to get the output of the round. This round function is applied sixteen times in total and is displayed in figure 3 below.

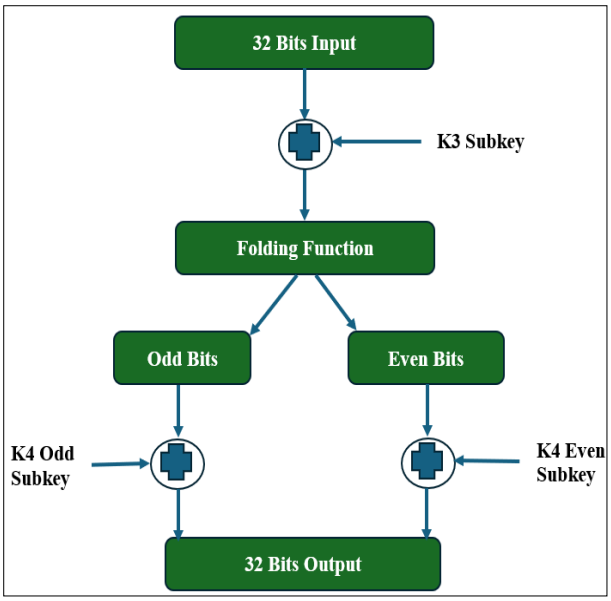

Fig 3. Round Function in Proposed Cipher

2.3 Swapping Function

The left and right halves of the output are exchanged. It's important to note that these operations are carried out exclusively on the 32-bit right half of the 64-bit input. The left half remains unchanged until this point. Now, the old right half becomes the new left half, and the old left half becomes the new right half.

2.4 Inverse Substitution

The final stage in the encoding procedure is the inverse substitution which is performed using the substitution table which is also prepared from the table 1.

3. DECRYPTION PROCEDURE OF FBC

The decryption process in this algorithm is exactly the reverse of the encryption method as it is based on symmetric cryptography.

4. PERFORMANCE EVALUATION

The algorithm has been implemented in Java and it can be implemented in any language that supports Unicode systems like Python, C, C++, etc. The basic idea behind developing a cryptographic algorithm is defense provided to the data being transferred across an untrusted link[6]. The adaptation of any encryption algorithm depends upon the tradeoff between security and speed[7][8]. If the algorithm offers great speed with greater protection against unauthorized attacks, it is widely accepted. Cryptosystems developed on the concept of block encryption suffered from the problem of producing the same output for same input and same encoding key, which make them vulnerable to 'replay attacks' [9]. However, using alternative block-cipher mode like Cipher Feedback (CFB) or Cipher Block Chaining (CBC) will add an additional layer of security that doesn't suffer from replay attacks [10].

Apart from the enhanced security mechanisms, FBC algorithm is pretty fast because of code optimization techniques employed in the programming. Especially loop optimization is done because loops consume maximum time of any program. The cipher algorithms are implemented on Intel(R) Core(TM) i5-1035G1 CPU @ 1.19 GHz with 8 GB RAM and Windows 11 64-bit Operating System. To test the speed of the proposed algorithm, it is compared with the existing standards like Triple Data Encryption Standard (3DES), Advanced Encryption Standard (AES) and Blowfish with variable size input files. It is vital to know that the deciphering timings of FBC are similar to the enciphering timings. The encryption timings (in seconds) for the FBC cryptosystem and renowned existing standards have been observed and listed in the following table II.

Table II: Speed Comparison of FBC and Existing Standards

File Size (Bytes)	3DES	AES	Blowfish	FBC
36000	2	4	1.5	2
60000	4	6	3	3
160000	10	15	8	8
230000	15	22	12	12

The comparison analysis of Feistel Based Cipher (FBC), 3DES, AES and Blowfish can be visualized using the following figure 4.

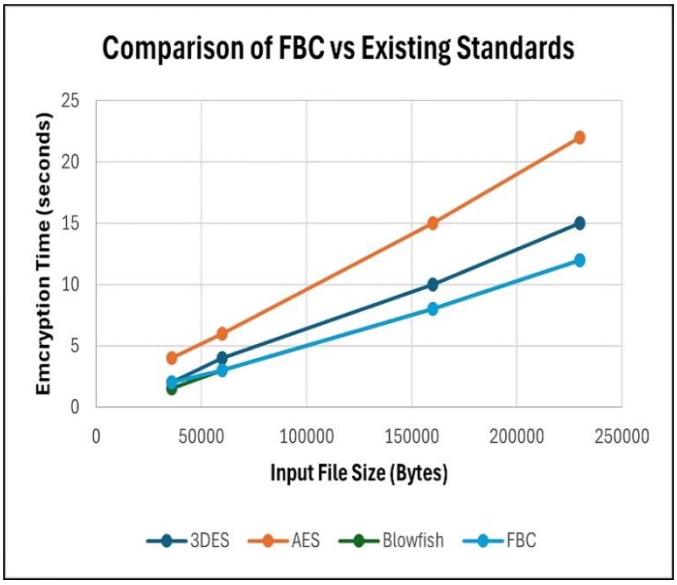

Fig 4. Comparison of Encryption Timings of FBC with Existing Standards

As demonstrated through figure above, the enciphering timings of FBC is much lower than the 3DES and AES, while the encoding timings of FBC is quite closer to the Blowfish algorithm. The security in FBC is also done on numerous levels as the key is divided into subkeys and applied on various levels, which further enhances the safeguarding of crucial data. The following figure (fig. 5)

illustrates the sample plaintext and corresponding ciphertext of FBC encryption algorithm.:

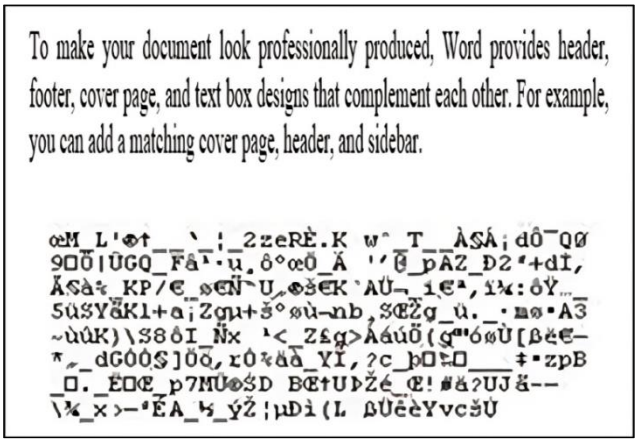

Fig 5. The Input Plaintext & its Corresponidng Ciphertext of the Proposed Cipher

5. CONCLUSION

The algorithm proposed in this paper is Feistel Based Cryptosystem (FBC) is based on the concept of Feistel structure. The algorithm encompasses 16 rounds and each round is further encrypted using XOR, folding function, Even-Odd function and swapping function. The cipher additionally uses the initial substitution and inverse substitution. All these extra layers of security ensure the proposed cipher provides the much-needed robustness against unauthorized attacks. The enciphering and deciphering timings of planned cipher also demonstrate that it is better and faster than the existing standards. FBC efficacy in text data encryption along with its supreme security; seems like an answer to the future encryption issues.

Future development will include:

- Implementation of FBC for image and audio data.
- Hardware realization of FBC.
- Usage of compression technique along with encryption procedure.

6. REFERENCES

[1] J. F. Dooley, "History of cryptography and cryptanalysis, History of Computing", 2018.

[2] "Basic Cryptographic Algorithms", an article available atwww.itsc.state.md.us/oldsite/info/internetSecurity/Crypto/Cry-ptoIntro.htm#Algorithms.

[3] A. Biryukov, G. Leurent, and L. Perrin, "Cryptanalysis of Feistel networks with secret round functions", In International Conference on Selected Areas in Cryptography (pp. 102-121), Cham: Springer International Publishing, August, 2015.

[4] D. Bujari, and A. Erke, "Comparative analysis of block cipher modes of operation", In International Advanced Researches & Engineering Congress, 2017.

[5] N. Kaaniche, and M. Laurent, "Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms", Computer Communications, 111, pp. 120-141, 2017.

[6] B. Bhushan, G. Sahoo, and A. K. Rai, "Man-in-the-middle Attack in Wireless and Computer Networking—A Review", In 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA), pp. 1-6, 2017.

[7] M. Usman, I. Ahmed, M. I. Aslam, S. Khan, and U. A. Shah, "SIT: a lightweight encryption algorithm for secure internet of things", arXiv preprint arXiv:1704.08688, 2017.

[8] S.R. Kimbleton & G.M. Schneider, "Computer communications networks: approaches, objectives and performance considerations", ibid., pp. 129-173, 1975.

[9] A. Hoehn, and P. Zhang, "Detection of replay attacks in cyber-physical systems", In 2016 American control conference (ACC), pp. 290-295, 2016.

[10] V. K. Pachghare, "Cryptography and Information Security", PHI Learning Pvt. Ltd, 2019.