TEST Engineering & Management

September-October 2019
Page No.10 - 16

Novel Encryption Algorithm for Securing Sensitive Information Based on Feistel Cipher

Shams Mahmoud Abd Ali[1,a], Hasan Falah Hasan [2,b]
[1]Translation Department, College of Art, Al-Iraqia University, Iraq
[2]Computer Engineering Department, College of Engineering, Al-Iraqia University, Iraq

Article Info
Page Number: 10 - 16
Publication Issue:
September-October 2019

Article History
Article Received: *12 January 2019*
Revised: *02 April 2019*
Accepted: *05 August 2019*
Publication: *20 October 2019*

Abstract: The growth of latest technology specially internet and its applications imposes a major concern in respect to information security in terms of protecting precious content against any possible and vital attacks. Cryptography has made the difference over period of time by using developed encryption algorithms where some of them have successfully fulfill its designed goals however, others have failed due to specific requirements or encryption/decryption procedures have been recognized. Recent encryption algorithms employed aims mainly to escalate the security levels of sensitive information in order to deliver confidentiality, authentication, integrity, and non-repudiation. In this paper, the proposed encryption algorithm has combined the core of three different algorithms DES, Blowfish and Vigenère but with optimized secure. The key has been enhanced to be 256 bit key inspired by AES algorithm key and Blowfish PI factor. Key generation and encryption/decryption performing using feistel cipher methodology. Analysis and experimental results have clearly proven solid randomness on key generation, Encryption and decryption in which it has met all targets designed.

Keywords: *Cryptography, Encryption, Information Security, DES, Blowfish, AES, Vigenère, Feistel cipher.*

1. Introduction:

Sensitive Information have been exchanged over various medium constituting revolutionary jump among benefited parties in different business aspects. They have conducted a major concern for the owners due to them contents. Medical, financial, commercial..etc information demand an authentic and secure environment in order to be protected. It asserts the fact of need to comprehensive and secure mode to increase confidentiality and authenticity.

Cryptography is the notion of protecting confidential data from unauthorized access in order to overcome possible attacks. It transforms readable to a non-readable data format. Encryption and decryption concepts rise as a core engine of cryptography in which, encryption imposes such methodology to mutate readable information to a ciphered format. Decryption is the restoring process of original text however, the act of data encryption can be used by all user levels. Encryption and decryption falls into two major models symmetric and asymmetric. Symmetric key encryption model built algorithms using identical key in which, it is playing a role of maintaining shared secretes among parties however sharing the same key in the previous model is considered as a major drawback due to the possible ability of revealing ciphered text [1,7,11]. Symmetric key encryption can use either stream ciphers or block ciphers. Stream ciphers encrypt one digit at a time of a certain text. Block ciphers take a number of bits and encrypt them as a single unit. Blocks of 64 bits have been commonly used. Asymmetric encryption

model is used to solve key distribution where it focuses on generating two types of keys. Public key which is known to public in order to encrypt plain text and private key to decrypt ciphered text acquired by designated user. A Feistel structure implements a series of iterative ciphers on a block of data and is generally designed for block ciphers that encrypt large quantities of data [4,8,11]. It works by splitting the data block into two equal pieces and applying encryption in multiple rounds. Each round implements permutation and combinations derived from the primary function or key. The number of rounds varies for each cipher that implements a Feistel structure. Moreover, if the algorithm used in reversible mode, it produces the same output until the input is the same. Data Encryption Standards (DES) algorithm was developed to provide standard mode of protecting data. It uses two basic techniques of cryptography confusion and diffusion. diffusion is achieved through several permutations and confusions achieved through the XOR operation [11,12]. Encryption starts with an initial permutation of the 64 input bits. These bits are then divided into two 32-bit halves (L and R). The encryption then proceeds through 16 rounds, each using the existing L and R parts, and a subkey. The R and subkeys are processed in a function f, and the outputs of the f function are exclusive-XORed with the existing L part to create the new R part. The new L part is simply a copy of the incoming R part. In the final round, the L and R parts are swapped once more before the final permutation, producing the output block. Decryption is identical to encryption, except that the subkeys are used in the opposite order. Key Strengths considered outlined as adaptability and simple core Feistel structure and even, it doesn't need vast processing power but to some extent key weaknesses are effecting the integrity of the algorithm processing due to small key size where susceptible to brute force attacks are highly possible [1,7,8,13].

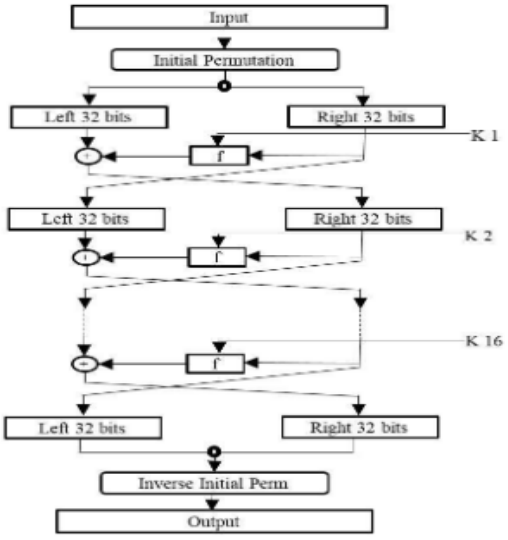

Figure 1 DES Algorithm

The AES / Rijndael cipher was the winner of the finalists selected to replace the formerly secure DES system. Unlike its predecessor, it does not use the Feistel network, but instead has a substitution - permutation structure. It has three key strengths (128, 192, 256) which have in turn a related number of repetitive rounds (10, 12, 14) in which the plaintext is transformed into the cipher text. The plaintext is broken up into 128-bits or 16 bytes which are formed in to 4x4 byte blocks, they are copied to the state array this is modified at each stage of encryption / decryption by the four functions: byte substitution, permutation, arithmetic operation over a finite field and XORed with key. These functions in the initial rounds involve substituting each byte, shifting to rows, mixing columns and adding the key [14,15]. The final round only has three of these functions. The process is illustrated below. The strength of the AES rely as, it has Larger key than DES and has fair key setup time, Efficient Performance / low memory requirements and Effective combination of functions for encryption / decryption[18,19].

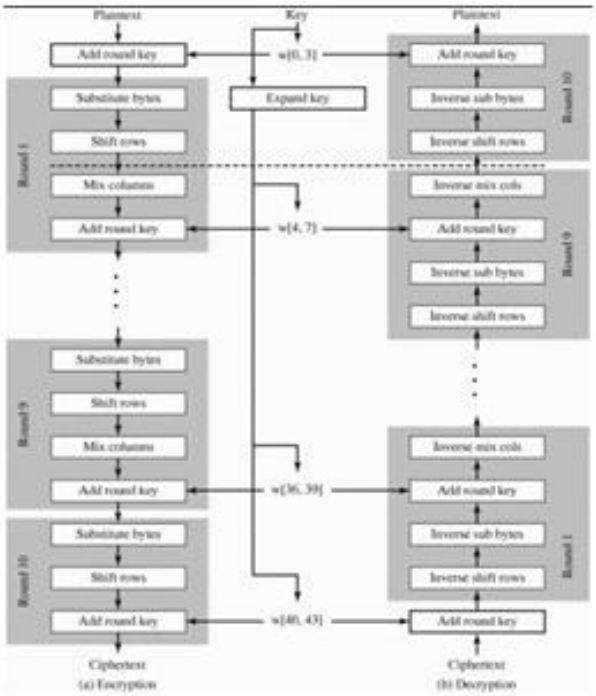

Figure 2 AES Algorithm

Blowfish is a symmetric block cipher originally developed as a replacement for DES. Its built is similar to DES, has a Feistel structure, but it has a key size of up to 448 bits rather than 56 bits, and the encryption (and decryption) process is also more complex. Blowfish is considered fairly secure; so far none have successfully found a way to decrypt it using cryptanalysis. Blowfish takes a 64-bit block of plaintext and encrypts it through a 16-round Feistel cipher that uses large key-dependant S boxes. A great strength of Blowfish lies in the way it generates these subkeys, Blowfish uses the decimal numbers of PI because they show no obvious pattern. The secret key will initially be XORed with decimal numbers from pi to create initial subkeys. These will then be regenerated using the Feistel cipher and the first subkeys, to end up with the final array of subkeys. It's this final array that will then be used to encrypt the plaintext [13-15]. Its Feistel network with F being the Feistel function, using only modular addition and XOR. It is unbreakable with cryptanalysis where key size reach up to 448 bit moreover, it is complex, hard to break by brute force attack and uses PI for key generation in order to avoid pattern recognition. The major drawback appointed to this algorithm is a resource intensive.

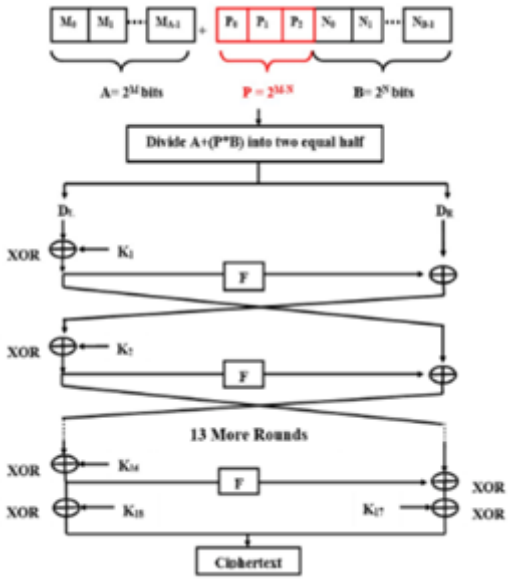

Figure 2 Blowfish Algorithm Encryption Procedure

Vigenère cipher given an adequate sized ciphertext, it is possibly through frequency analysis to uncover patterns and therefore letters in a given language, leading to the cipher being cracked and text decoded. One of the ways to suppress and dampen the frequency of letters in such a cipher is to use a polyalphabetic substitution cipher. The Vigenère cipher consists of a keyword that defines how the

plaintext should be looked up and encoded via a table called the Vigenère tableau where, each row corresponds to a Caesar Cipher shift of between 0 to 25[16,18]. It is Easy to execute, explain and generate cipher text and Requires little processing power, can be done by hand where its weaknesses can be decoded with little processing power and Simplistic, lack of numerous cycles and transformations. The proposed algorithm in this paper has bounded the core of the DES cipher but with optimized secure. The key size has been inspired from the AES to be a 256 bit key. While looking at Blowfish inspiration is presented too by the use of pi as it would be an effective device in key generation. Both key generation and encryption/decryption procedures follow the Feistel network. The reminder of this paper is arranged as follow, in section 2 contains key generation, encryption and decryption processes. Section 3 illustrating the experimental results via key generating either automatically or manually, encoding any input y user and decoding the coded output. Section 4 includes conclusion and future work.

2. Proposed Algorithm

2.1 Key Generation:

The proposed algorithm uses a 256-bit key, generated by hashing the textual password using the SHA-256 hashing function. The generated key will then produce a 16 sub key having 64 bit length in the P-array that will be used in the encryption and decryption Feistel cipher. In order to generate this P-array, the need to an initial P-array also holding 16 sub keys. The construction of this initial P-array by filling it up with the hexadecimal values from 256 decimals of PI. The decimals of PI will be used in order to increase key randomness however, that will make cryptanalysis harder. The key generation of the final 16 sub key P-array is done using a stripped down version of the Feistel cipher that's almost the same as the one used for encryption with the only difference being the amount of rounds each block goes through. Firstly the 256-bit key is divided into two parts, 128 bits each. Each of these parts then goes through an 8 round Feistel scheme. The result after every round is saved as a final sub key in the final P-array. Eight rounds will produce eight sub keys, but because there are two 128-bit blocks that will results 16 sub keys. Algorithm1 presents the key generation, figure 1 illustrates final subkey production.

Key Generation Algorithm:

1. Split the 128-bit block in two parts of 64-bit
2. Pass the second 64-bit part straight on to the next round
3. XOR the first 64 bits with the first sub key
4. The result is then substituted with the first sub key to get the final result. The substitution uses a 256x256 matrix, filled with values from 0 to 255. The way the matrix is filled is based on the Vigenère cipher.
5. This final result will be saved in the P-array as the first sub key.
6. Steps 2 to 5 are repeated in the next 7 rounds of the Feistel scheme, storing another sub key at each round.

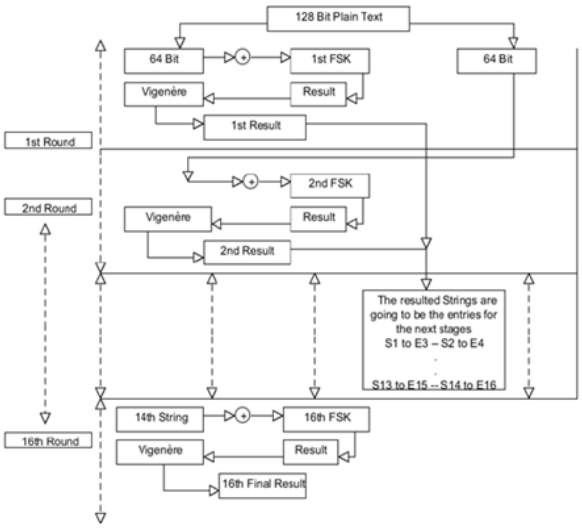

Figure1 Final Subkeys

The keys resulted from each round are stored in a separate array, and the process continues until full 16 sub keys have achieved. Figure 2 showing final subkeys storage.

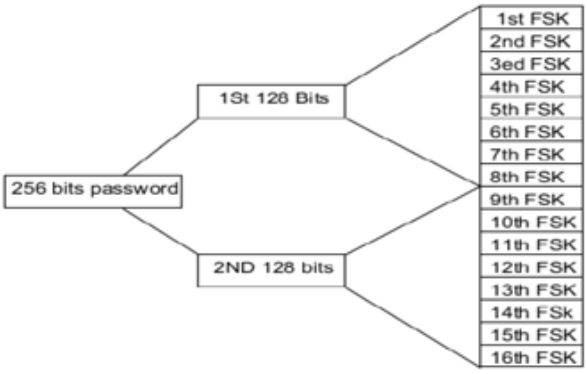

Figure 1 Final Subkeys Storage

2.2 Encryption Process

At first, the plain text is divided into 128-bit blocks. If the last block is less than 128 bits long, then it is padded with zeros until it reaches 128 bits. Each 128-bit block of the plaintext is then put through a 16 round Feistel scheme. The function of each round of the Feistel scheme is the same as those from the key generation, meaning first an XOR and secondly a substitution. The sub key used for each round will come from our P-array that we obtained after the key generation. This process will continue for every 128-bit block, resulting cipher text. The final cipher text obtained will be outputted using the Base64 character set, which means that every character will be printable and therefore easy to pass on. Figure 3 presents text ciphering procedure.

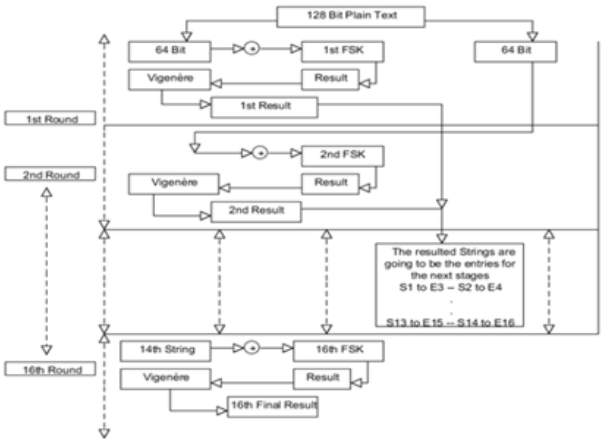

Figure 3 Plain Text Ciphering Mechanism

3. Analysis and Comparison

The proposed algorithm has employed previously illustrated algorithms cores however, it has its own perspective of performing encryption and decryption via modified either cores or size and key generation. One of the potential aims was to keep its core structure simple hence the adoption of aspects of the Feistel network for both its key generator and encryption/decryption process. This stable yet simple process allows the user to add further permutations to the process if so desired. Unlike DES it has been strengthen with a longer key, comparable to AES or even AES complicated core however, it utilizes the blowfish idea of PI which allows for more random keys and makes effective cryptanalysis harder. It is strengthened further by adding a substitution function, inspired by AES and the Vigenère cipher. The cipher requires little processing power, is flexible in nature, simple less resource consumption and easy to explain yet has a strong key generation process. Unlike others where, they are is a resource intensive. The level of security should be greater than that offered by DES and 3DES. While its transformation process is not as extensive as the AES algorithm (four key functions), it does have the same key size and two more rounds than the 256 key AES cipher. While its sixteen round network has been built to withstand known block cipher attacks, it is also adaptable enough to compensate for any computational developments in the future.

4. Experimental Results

The proposed algorithm opens up an applet upon execution, which defaults to the encoding tab, assuming the user wants to initially encode a block of plaintext. The user then types in the text to be encoded in the upper box and has a choice of either typing in their own key or clicking on the key generation button. Once done the user clicks the encode button to produce the cipher text as illustrated below:

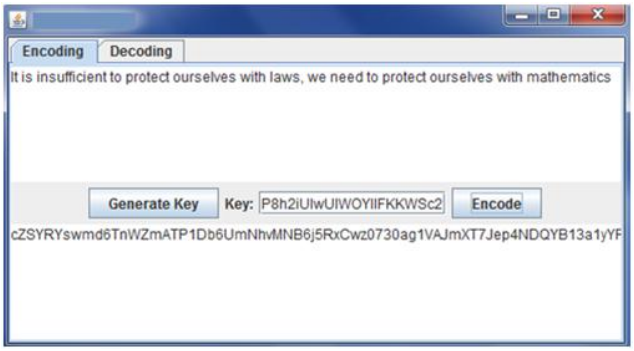

Figure 6 Encryption Scenario

To decode a block of cipher text the user clicks the decoding tab and copies in the cipher text into the upper box. They must ensure that they type the appropriate key into the space, once done the decoding process can be executed by clicking the button, thus giving the original plaintext.

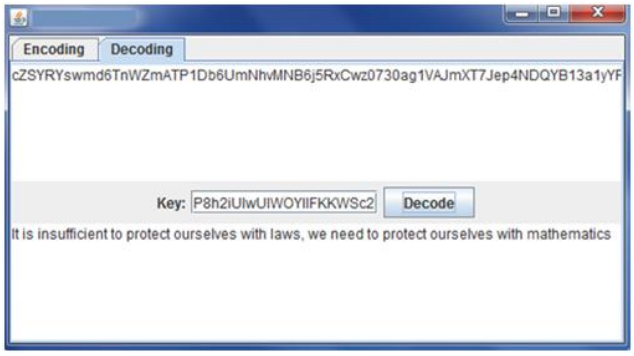

Figure7 Decryption Scenario

5. Conclusion and Future Work

Most of information exchanged considered sensitive due to the nature of technology has been used these days. Cryptography has been used since long time via different encryption and decryption algorithms through its symmetric and asymmetric manner but with different stands of performing methodology while, quite good number of these algorithms have been pattern recognized. In this paper, the proposed algorithm has employed the methodology of Feistel structure through using series of iterative ciphers on a block of data technique however, it has its own style of performing encryption and decryption via modified cores and keys size generation. User can add as much permutation to the process in order to harden either key generation or encryption required unlike other algorithm methodologies in which, they are focuses on strengthen either their keys or make encryption decryption more complex. It focuses on how to improve blowfish, AES and Vigenère features in terms of using PI and substitution that makes cryptanalysis more efficient and hard enough. Elastic nature, little resources consumption, and simple structure are the bases have been used to build the proposed algorithm but with strong key generation performance where, DES is not secure enough and AES are more complicated on generating its four level key. It built sixteen round network to withstand known block cipher attacks, it is also adaptable enough to compensate for any computational developments in the future. Potential future work can be made via enhancing the number of rounds to generate stronger encryption/decryption key.

References:

[1] Jarjar, A. (2018). Optik Improvement of Feistel method and the new encryption scheme. Optic - International Journal for Light and Electron Optics, 157, PP. 1319–1324

[2] Zhang, X., Zhou, Z., & Niu, Y. (2018). An Image Encryption Method Based on the Feistel Network and Dynamic DNA Encoding An Image Encryption Method Based on the Feistel Network and Dynamic. IEEE Photonics Journal, 10(4), PP. 1–14.

[3] L. Y. Zhang et al., "On the security of a class of diffusion mechanisms for image encryption," IEEE Trans. Cybernetics., vol. 48, no. 4, pp. 1–13, Apr. 2017.

[4] Zhang, X., Zhou, Z., & Niu, Y. (2018). An Image Encryption Method Based on the Feistel Network and Dynamic DNA Encoding An Image Encryption Method Based on the Feistel Network and Dynamic. IEEE Photonics Journal, 10(4), 1–14.

[5] Patel, R., & Kamboj, P. (2017). Security Enhancement of Blowfish Block Cipher, Springer, PP. 231–238.

[6] Fathima, H., Matriculation, K. S. R., & Kalvi, K. S. R. (2017). Comparative Study of Symmetric Key. Global Journal of Computer Science and Technology, 17(2), PP. 13–16.

[7] Baker, S. I. B. (2017). Novel Algorithm in Symmetric Encryption (NASE). IEEE International Conference on New Trends in Computing Sciences, (3), PP.191–196.

[8] Cruz, B. F., Domingo, K. N., Guzman, F. E. De, Cotiangco, J. B., & Hilario, C. B. (2017). Expanded 128-bit Data Encryption Standard. International Journal of Computer Science and Mobile Computing, 6(8), 133–142.

[9] Patel, P., Patel, R., & Patel, N. (2016). Integrated ECC and Blowfish for Smartphone Security. Procedia - Procedia Computer Science, 78, PP. 210–216.

[10] Mahamat, Y., Othman, S. H., Siraj, M., & Nkiama, H. (2016). Comparative Study Of AES , Blowfish , CAST-128 And DES Encryption Algorithm International

organization of Scientific Research. IOSR Journal of Engineering, 6(6), PP. 1–7.

[11] Y. Erlich andD. Zielinski, "DNA fountain enables a robust and efficient storage architecture," Science, vol. 355, no. 6328, pp. 950–954, 2016.

[12] Rejani, R., & Krishnan, D. V. (2015). Study of Symmetric key Cryptography Algorithms. International Journal of Computer Techniques, 2(2), 45–50.

[13] Bansal, V. P. (2015). A Hybrid Data Encryption Technique using RSA and Blowfish for Cloud Computing on FPGAs. IEEE, Proceeding. PP.95-104.

[14] Survey, R. C. A., & Princy, P. (2015). A COMPARISON OF SYMMETRIC KEY ALGORITHMS DES , AES , BLOWFISH ,. International Journal of Computer Science & Engineering Technology, 6(5), PP. 328–331.

[15] Peng, J., Lei, L., Han, Q., & Jia, R. (2014). A Chaos-based Block Cipher with Feistel Structure. IEEE Conference on Cognitive Informatics & Cognitive Computing, PP. 343–348.

[16] Gupta, A., & Walia, N. K. (2014). Cryptography Algorithms : A Review. International Journal of Engineering Development and Research, 2(2), PP. 1667–1672.

[17] Srikantaswamy, S. G. (2013). A New Approach for Designing Cryptographic Systems based on Feistel Structure. International Journal on Computer Science and Engineering, 5(1), PP. 37–42.

[18] Aggarwal, K. (2013). Performance Evaluation of RC6 , Blowfish , DES , IDEA , CAST-128 Block Ciphers. International Journal of Computer Science Applications, 68(25), PP. 10–16.

[19] Tadepalli, G. (2014). A Novel Approach for New MBK-128 Cryptographic Algorithm A Novel Approach for New MBK-128 Cryptographic Algorithm. WULFENIA Journal, 19, PP. 90–100.