IJCTA, 9(10), 2016, pp. 4761-4769
© International Science Press

A Hybrid Blowfish Encryption Algorithm Using Nash Equilibrium with Cautions Attackers

***V. Josephraj **B. Shamina Ross**

Abstract : Cryptography may be described as the science of secure communication over a public channel. Cryptography algorithm is the technique that makes data or network secure by providing security. Cryptographic algorithm is classified into two categories. They are symmetric key cryptography and asymmetric key cryptography. An important area of cryptography is the symmetric cryptosystem, in which the encryption and the decryption keys are the same. Among symmetric cryptosystems, ciphers of different security levels have been developed, ranging from the substitution and transposition ciphers to block ciphers, such as Blowfish. So far, Blowfish has no cryptanalysis. This paper proposed a new hybrid Blowfish encryption algorithm combining Blowfish and the Game Theory's cautious attackers in Nash Equilibrium (NE) state, named as NE-Blowfish to improve the performance of the Blowfish cryptography algorithm by making modifications to the Feistel (F) function. The outcome of the Blowfish and NE-Blowfish algorithms are compared using Avalanche Effect to show the better performance of NE-Blowfish in terms of security.

Keywords : Avalanche Effect, Blowfish, Cryptanalysis, Feistel Network, Nash Equilibrium

1. INTRODUCTION

Electronic security is becoming very important day by day as the internet and other forms of electronic communication are becoming more popular. Cryptography plays an important role for protecting data from devastating forces and intruders. Cryptographic algorithms have mathematically become more and more complex with time due to the ever increasing need for data security. The increase in the complexity of such algorithms requires more computation, execution time and high energy consumption. Successful studies have been made to speed up the execution of cryptographic algorithms. On the basis of input data, cipher algorithms are classified as block ciphers and stream ciphers. The Blowfish algorithm was designed by Bruce Schneier to replace Data Encryption Standard [1]. It is a symmetrical block cipher [2] having the advantages of secure, fast, easy to implement etc. Symmetric ciphers require less computational processing power than asymmetric ciphers. Symmetric ciphers are about 1000 times faster than asymmetric ciphers [3]. The operation part of Blowfish consists of XORs and additions on 32-bit words, and only 4KB or even less memory is needed when it runs. The key length of Blowfish is anywhere from 32 bits to 448 bits, which makes datum safe enough. The proposed hybrid NE-Blowfish algorithm obtained by combining Nash Equilibrium and Parallel Blowfish algorithms enhances the performance of Blowfish algorithm by modifying the function F of the existing Blowfish algorithm. There are a lot of benefits from parallel computing. The advantage of this system is its ability to handle large and extremely complex computations. According to Amdhal's law, if a problem can be decomposed and parallelized to run on multiple cores simultaneously, then there will be a speed gain [4].The Blowfish and Nash Equilibrium concept in Game Theory are combined so that

* Department of Computer Science Kamaraj College Manonmaniam Sundaranar University Thoothukudi-628003, India. Email: v.jose08@gmail.com Mobile: 00919443151625

** Department of Computer Applications Scott Christian College Manonmaniam Sundaranar University Nagercoil-629003, India. Email: shaminas@hotmail.com Mobile: 00919443137232

the security is increased. The Avalanche effect is used to show that the proposed hybrid NE-Blowfish algorithm possess good diffusion characteristics as that of original Blowfish algorithm [5]. The objective of this research paper is to study the Blowfish algorithm and improve its performance using Parallel Processing and Game Theory based Nash Equilibrium techniques with cautious attackers.

2. RELATED WORK

A. Game Theory

The field of game theory and cryptographic protocol design are both concerned with the study of interactions among mutually distrusting parties. Players are assumed to be rational in Game Theoretic settings. To capture the nature of rational behavior, a great deal of effort was invested, resulting in a long line of stability concepts. Assuming that some parties are honest and follow the protocol, while some parties are malicious and behave in an arbitrary fashion, the cryptographic protocols are designed. Game Theory studies interactions between players with the same or conflicting interests. Game Theory plays a vital role in Logic and Computer Science. For modeling interactive computations, Games are used by the scientists. The use of algorithms for finding equilibria in games was initiated only after internet came into existence. In recent days Game Theory finds extensive application in network security.

B. Nash Equilibrium

In game theory the term Nash Equilibrium is used to describe equilibrium where each player's strategy is the most favourable given the strategies of all other players. A Nash Equilibrium exists when there is no one side profitable move from any of the players involved. No player in the game would take a different move as long as every other player's state remains unaltered. When players are at a Nash Equilibrium they do not have the intent to move because they will be in worse situation since Nash Equilibria are self-enforcing. Assuming that other players remain constant in their strategies, an individual can receive no incremental benefit from changing actions. A game may have either one or more Nash equilibria or none at all. The unique Nash equilibrium of the game is where both the players concede. If neither of them conceded, both would be better positioned. Nash equilibrium is defined as a stable state in which no player can do better than the other. Assume a situation in which both the players choose not to concede. In such a situation, given that Player B is not conceding, Player A would be better positioned by moving back and choosing to concede instead. There are various types of equilibrium in Game Theory. Of all, Nash equilibrium is the best. In Nash equilibrium the strategies of players are independent.

C. Parallel Processing

Parallel processing is the process of running a program in less time by dividing the program instructions among multiple processors. All the processors work in the same way. Based on the instructions written in assembly language for these processors, they perform mathematical operations retrieving the data from computer memory. The processors can also move data to a different memory location. They use software to communicate information to other processors. The processors can adjust data values and stay in sync with one another by exchanging messages. Once all the processors finish their tasks, the CPU reassembles all the individual solutions into an overall solution for the original computational problem. Latency and bandwidth are the two major factors that can impact system performance. The amount of time taken by the processor to transmit results back to the system is referred to as latency. It is not good if the processor takes less time to run an algorithm than it does to transmit the resulting information back to the overall system. In such cases, a sequential computer system would be more appropriate. Bandwidth refers to the amount of data the processor can transmit in a specific span of time. For a good parallel processing system the latency will be low and the bandwidth will be high.

3. BLOWFISH ALGORITHM

The Blowfish algorithm inputs a 64-bit plaintext and then outputs a 64-bit cipher text. It uses a variable length key of size from 32 bits to 448 bits [6], making it ideal for both domestic and exportable use. The Feistel structure of Blowfish algorithm is shown in Figure 1

There are two parts in the algorithm. One is the key expansion part and the other is the data encryption part. Key expansion converts a key of maximum length 448 bits into several sub key arrays to a total of 4168 bytes. The original sub key p-box and s-box are fixed. A fixed string consisting of hexadecimal digits of Pi (less the initial 3) is used to initialize them in order. Data encryption takes place via a 16-round Feistel network [7] after key expansion. The algorithm uses two boxes, a key dependent permutation key *p*-box [18] and data dependent substitution key *s*-box [4] [256], and a core Feistel function. The two boxes require 18 × 32 + 256 × 32 = 4186 bytes memory. The sub keys must be pre-computed before any data encryption or decryption. Function F is obtained by dividing XL into four eight-bit quarters, *a*, *b*, *c*, and *d*

$$F (XL) = ((S1, a + S2, b \bmod 2\text{^}32) \text{ XOR } S3,c) + S4, d \bmod 2\text{^}32$$

In the above F function, "+" is addition on 32-bit words, and XOR represents Exclusive OR. S1, a represents key *s*-box [1] [*a*], S2, *b* represents key s-box [2] [*b*], S3, *c* represents key s-box [3] [*c*], and S4, d represents key *s*-box [4] [*d*]. The key p-box is used in the reverse order for decryption process. Figure 2 shows the calculation of the function F(XL) using Blowfish algorithm.

The principle of Blowfish algorithm is both easy to understand and easy to implement. Different with other ciphers, all sub keys of Blowfish are influenced by every bit of the key, that makes the key and the data mingled together completely, which makes it quite difficult to analyze the key. The function F gives the Feistel network a great avalanche effect.

Blowfish algorithm finds extensive application in the field of information security as it is not only secure, but also fast, and suitable for different platforms [8]. Blowfish is among the fastest block ciphers available [9]. Blowfish is used in wide range of applications such as bulk encryption of data files, remote backup of hard disk. Also multimedia applications use blowfish for encryption of voice and media files. It is now being used in biometric identification and authentication, using voice, facial or fingerprint recognition. Geographical information system uses blowfish for cryptographic protection of sensitive data. These applications run in high-end servers, workstations, process bulk amount of data and demand high speed encryption and higher throughput [10]. Blowfish had a very good performance compared to the algorithms DES, 3DES and AES when tested on two different hardware platforms with input files of varying contents and sizes. [11]. Bruce Schneier made a block cipher speed comparison among Blowfish, RC5, DES, IDEA, 3DES algorithms. The Blowfish algorithm took nine clock cycles for each round whereas RC5 took 12 clock cycles per round, DES and 3DES took 18 clock cycles per round and IDEA 50 clock cycles per round. Blowfish, RC5, and DES algorithms perform 16 rounds whereas IDEA 8 rounds and 3DES 48 rounds of iterations. Of all the algorithms discussed, Blowfish has the least number of clock cycle per byte of output. The results showed the better performance of Blowfish among block ciphers in terms of speed. The results are shown in Table 1. From the Table 1 it is clear that, the Blowfish has a promising future.

4. PROPOSED NASH EQUILIBRIUM BLOWFISH ALGORITHM AND ANALYSIS

A. NE-Blowfish

The block diagram representing the structure of NE-Blowfish algorithm is shown in Figure 3. The proposed NE-Blowfish is obtained by combining Parallel processing, Blowfish and Game Theory's Nash Equilibrium.

Paper [12] shows that the Game Theory Based Identification Model (GBIM) has atleast one Nash Equilibrium. For this research, the case where the two types of attackers adopt different strategies is considered. Here, the cautious attackers gain more benefit in Nash Equilibrium. In practice the cautious attackers are more inclined to adopt low frequency attack strategy. The Nash equilibrium for this case is the prime choice of the verifier. The proposed NE-Blowfish algorithm is obtained by incorporating the Nash Equilibrium with two attackers adopting different strategies. Parallel processing was not introduced in the original Blowfish algorithm. As the speed of the algorithm can be increased by parallel processing, the F function of the Blowfish algorithm is modified in such a way that parallel processing can be done and the Game Theory's Nash Equilibrium concept is also added without violating the security requirements. The F function of the proposed hybrid algorithm is obtained as follows

$$F(XL) = (S1, a\text{-}S2, b)/((S1, a\text{-}S2, b) + (S3, c\text{-}S4, d))$$

This modification supports the parallel evaluation of three subtraction operations, then one addition operation and finally a division operation. All these operations take place in 3 steps. Figure 4 shows the calculation of F function using NE-Blowfish.

B. Performance Comparisons

In this paper the performance metrics execution time, encryption time, decryption time, throughput, avalanche effect, power consumption are used to evaluate the blowfish algorithm and NE-Blowfish algorithm. The encryption time, the decryption time, the Execution time, is low for Blowfish algorithm than NE-Blowfish algorithm. But the Avalanche Effect is high for NE-Blowfish than Blowfish. NE-Blowfish is the best in terms of security. Blowfish algorithm by itself is highly secure. But above all NE-Blowfish is unbreakable in any circumstances.

5. EXPERIMENTAL RESULTS

For the purpose of this research a Laptop with Intel Pentium T4500 @ 2.30GHz CPU, 4.00GB Dual-Channel DDR3 and Linux Mint 17.1 was used to collect the performance data. The software encrypts the text file size that ranges from 50 bytes to 208942 bytes. The performance metrics are the avalanche effect, encryption speed, decryption speed, execution time, encryption throughput, decryption throughput and execution throughput. The NE-Blowfish cryptosystem was implemented using the C programming language in gcc compiler.

A. Avalanche Effect

When a change is made to one bit of the plain text or one bit of the key, there will be a considerable change in the cipher text. The number of bits changed in the cipher text is called Avalanche Effect [13]. For a cryptographic algorithm to be secure it should exhibit strong Avalanche effect. For this research one bit of the plain text is changed and the cipher text is examined for twenty times and the average of the twenty avalanche values is taken and tabulated. Tabulation of results observed by changing one bit of plain text in the sample is shown in Table 2. Figure 5 represents the Avalanche effect of Blowfish algorithm and NE-Blowfish algorithm. In the bar chart Blowfish is represented as BF and NE-Blowfish as NEBF. The Blowfish algorithm has the lowest Avalanche effect when compared to the hybrid NE-Blowfish algorithm discussed here. So it is clear that NE-Blowfish algorithm is more secure than Blowfish algorithm.

B. Encryption Time

The amount of time required for converting plaintext message to cipher text during the encryption process is the Encryption Time. Tabulation of results of the performance metric encryption time with fifteen different packet sizes ranging from 50 bytes to 208942 bytes for Blowfish algorithm is shown in Table 3 and for NE-Blowfish algorithm in Table 4. The encryption time of NE-Blowfish algorithm is slightly more than Blowfish algorithm.

C. Decryption Time

Decryption Time is one of the performance metrics which is defined as the amount of time required for converting the cipher text into the plain text at the time of decryption Tabulation of results of decryption time with fifteen different packet sizes ranging from 50 bytes to 208942 bytes for Blowfish algorithm is shown in Table 3 and for NE Blowfish algorithm in Table 4. The decryption time for NE-Blowfish algorithm is slightly more than Blowfish algorithm.

D. Execution Time

Execution time of an algorithm directly depends on the functionality of the algorithm and it clearly defines that more complex structure originates poor execution time. Higher the key length provides higher security but increases execution time. The speed of the algorithm is determined by the execution time of the algorithm. Tabulation of results of execution time with fifteen different packet sizes ranging from 50 bytes to 208942 bytes for Blowfish algorithm is shown in Table 3 and for NE-Blowfish algorithm in Table 4. The execution time for NE-Blowfish algorithm is slightly more than Blowfish algorithm which is reasonable.

E. Throughput

The Encryption throughput is obtained by dividing the total plaintext encrypted in Megabytes by the total encryption time in seconds for each algorithm.

Throughput = Total Plaintext in Mega Bytes / Encryption Time

The higher the value of throughput more is the efficiency of encrypting any text with an encryption algorithm.

The Decryption throughput is calculated by dividing the total plaintext in Megabytes decrypted by the total decryption time for each algorithm. The Execution throughput is calculated by dividing the total plaintext in Megabytes executed by the total execution time for each algorithm. Tabulation of results of encryption throughput, decryption throughput and execution throughput with fifteen different packet sizes ranging from 50 bytes to 208942 bytes for Blowfish algorithm and NE-Blowfish algorithm are shown in Table 3 and Table 4 for Blowfish and NE-Blowfish algorithms respectively.

Table 1. Block Cipher Speed Comparison

Algorithm	*Clocks/round*	*No. of Rounds*	*Clocks/byte of output*
Blowfish	9	16	18
RC5	12	16	23
DES	18	16	45
IDEA	50	8	50
3DES	18	48	108

Table 2. Avalanche Effect

Algorithm	*BF*	*NEBF*
Avalanche	57.1	67.2

Table 3. Speed Analysis of Blowfish Algorithm

Data size in Bytes	*Encryption*	*Decryption*	*Execution*
50	0.7586	0.7602	0.8875
60	0.7709	0.7722	0.9058
100	0.7919	0.7934	0.9543
250	0.8962	0.8978	1.1592
325	0.9486	0.9497	1.2615
700	1.2776	1.2005	1.7646
900	1.3354	1.3364	2.0352
965	1.3741	1.549	2.29
5350	4.5246	4.4654	8.3683
7400	5.9181	5.8499	11.146
9000	6.9128	5.1447	11.4318
51202	20.9473	16.2216	36.5376
61442	23.8123	19.2313	42.4173
102402	37.7555	31.5148	68.651
208942	63.2736	63.159	126.085
Average Time (millisec)	11.41983333	10.25639333	21.05967333
Throughput (MB/sec)	2.500233162	2.783848579	1.3557782

Table 4 Speed Analysis of NE-Blowfish Algorithm

Data size in Bytes	*Encryption*	*Decryption*	*Execution*
50	1.0371	1.0391	1.1786
60	1.0472	1.0494	1.199
100	1.0851	1.0984	1.274
250	1.2221	1.231	1.5494
325	1.2916	1.2941	1.6865
700	1.6394	1.7456	2.4931
900	1.8241	1.9301	2.8663
965	1.8826	2.0306	3.0228
5350	5.6879	3.8389	8.6176
7400	7.1893	4.4594	10.7308
9000	7.8691	5.0955	12.0576
51202	26.3078	22.172	47.5995
61442	30.8877	26.3397	56.3545
102402	45.3601	43.3051	87.7789
208942	87.2847	87.06862	173.9277
Average Time (millisec)	14.77438667	13.5798347	27.48908667
Throughput (MB/sec)	1.932550341	2.10254739	1.038675688

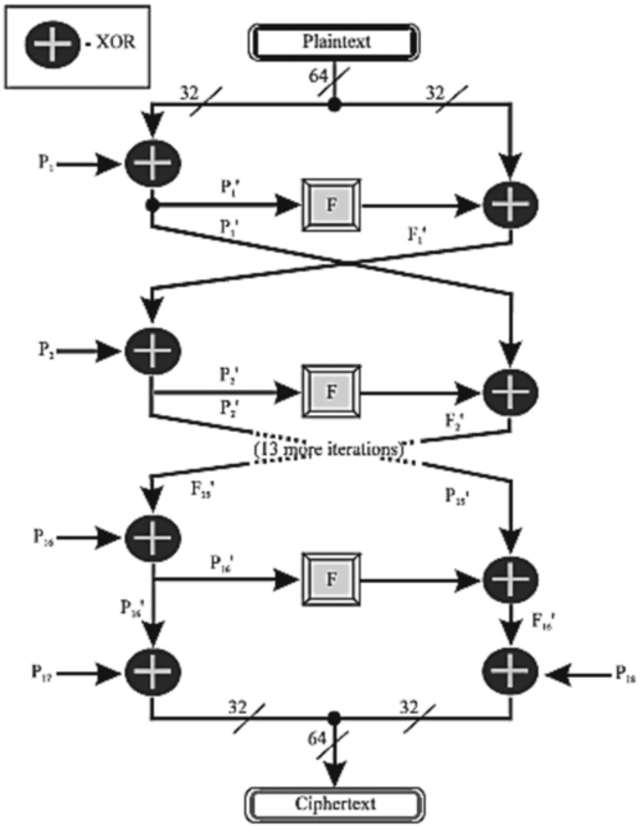

Fig. 1. Feistel Structure of Blowfish Algorithm.

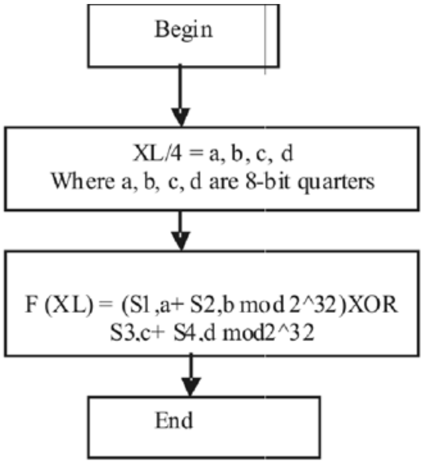
Begin
XL/4 = a, b, c, d
Where a, b, c, d are 8-bit quarters
F (XL) = (S1,a+ S2,b mod 2^32)XOR S3,c+ S4,d mod2^32
End

Fig. 2. F Function in Blowfish Algorithm

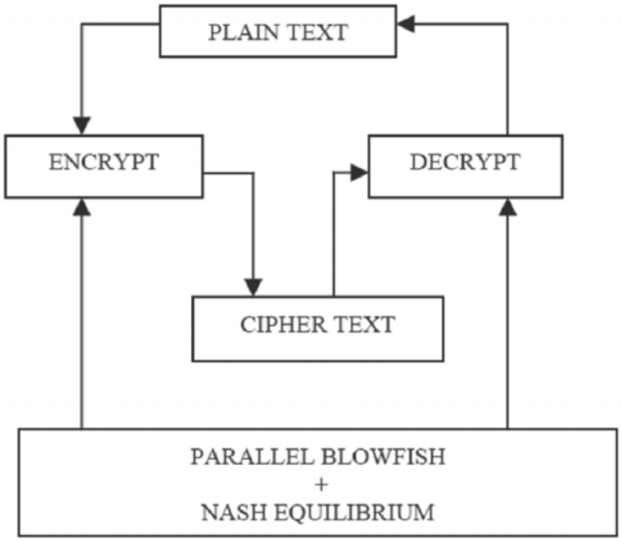

Fig. 3. Structure of NE-Blowfish Algorithm

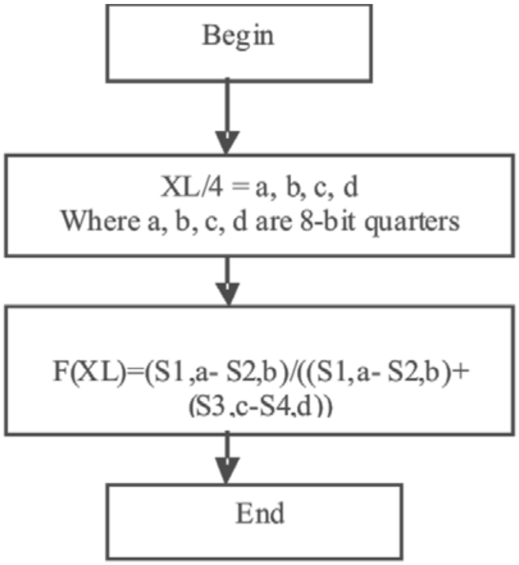
Begin
XL/4 = a, b, c, d
Where a, b, c, d are 8-bit quarters
F(XL)=(S1,a- S2,b)/((S1,a- S2,b)+ (S3,c-S4,d))
End

Fig. 4. F Function in NE-Blowfish Algorithm

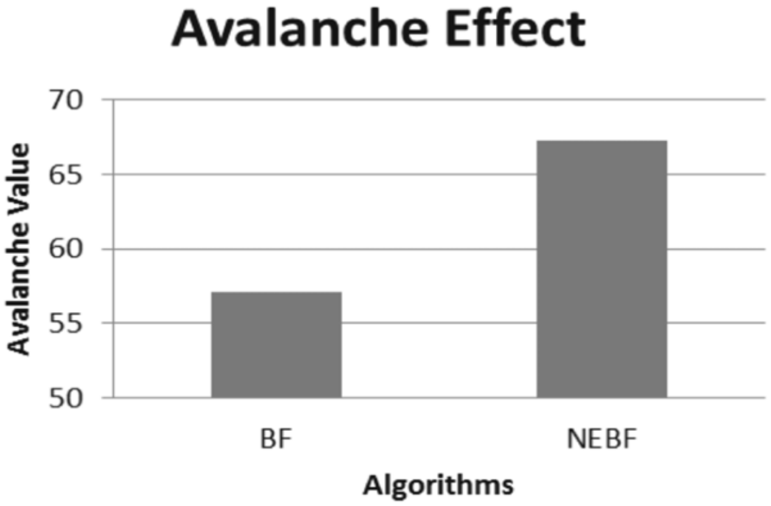

Fig. 5. Comparison of Avalanche Effect.

6. CONCLUSIONS

This paper gives a detailed study of the most popular symmetric key encryption algorithm that is Blowfish and discusses about its advantages. Based on the benefits and the bottlenecks of Blowfish algorithm, a new approach is proposed and implemented to further enhance the existing algorithm to achieve better results in terms of security. At each time the cipher text produced for the same input plaintext, will be different for both the Blowfish and the proposed hybrid NE-Blowfish algorithm. This is because every time a new random number gets generated and this as a result gives difference in the application of F function over each round. This enhances the security. The above results clearly indicate that the Avalanche effect of the new hybrid NE-Blowfish is much better than Blowfish algorithm. So it is clear that the hybrid NE-Blowfish algorithm obtained by combining the Blowfish, Parallel processing, Game Theory's Nash Equilibrium with cautious attackers is very strong, secure and unbreakable than the Blowfish algorithm. Hybrid NE-Blowfish can be widely used in PDAs and smart phones that have power, processor, and memory limitations. The research work can be further extended with other optimization techniques which have potential capacities.

7. REFERENCES

1. U.S. National Bureau of Standards, "Data encryption standard", U.S. Fed. Inform. Processing Standards Pub., FIPS PUB 46, pp. 2-27, January 1977..
2. Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd Edition, New York, John Wiley and Sons, Inc., pp. 21-27, 1996.
3. Selin Chandra C, Sujin Lal S, Saranya, "Evolution of Cryptographic Algorithms and Performance Parameters", "IIR Journal of Scientific Research Volume: 01 Issue: 01, pp. 18-23 June 2016.
4. Gene M. Adahl, "Validity of the single processor approach to large scale computing capabilities", in proceedings of the January, , spring joint computer conference AFIPS'67(Spring) ACM, Newyork, NY, USA, pp.483-485, January 1967
5. William Stallings, "Cryptography and Network Security", Fifth Edition, Pearson Education, pp. 2011,
6. Bruce Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)", in Cambridge Security Workshop on Fast Software Encryption, Cambridge, UK, December 9-11, 1993, pp.191-204.
7. Bruce.Schneier, "The Blowfish Encryption Algorithm", Dr. Dobb's Journal, Vol. 19, No. 4, pp.38-40, April 1994,
8. Mingyan Wang, Yanwen Que, "The Design and Implementation of Password Management System Based on Blowfish Cryptographic Algorithm", IEEE Xplore, International Forum on Computer Science- Technology and Applicatioins, IEEE Computer Society, 978-0-7695-3930-0/09, pp. 24-28, 2009.

9. B. Schneier, Speed Comparisons of Block Ciphers on a Pentium, retrieved, from <http://www.schneier.com/blowfish-speed.html>, August 27, 2010.

10. T. Srikanthan et al., Drill – A Flexible Architecture for Blowfish Encryption Using Dynamic Reconfiguration, Replication, Inner-Loop, Pipelining, Loop Folding Techniques", Springer- Verlag Berlin Heidelberg, pp. 6256-639, 2005.

11. Aamer Nadeem, Dr.M.Younus Javed, "A Performance Comparison of Data Encryption Algorithms", IEEE, Information and Communication Technologies, ICICT 2005, First International Conference, pp. 84-89, 2005.

12. J. Chen, Q. Yuan, G. Xue, R. Du, "Game-theory-based batch identification of invalid signatures in wireless mobile networks", IEEE Conference on Computer Communications (INFOCOM), Kowloon, pp. 262-270, 2015.

13. Krishnamurthy G. N., V.Ramaswamy, Leela G. H., Ashalatha M. E., "Performance enhancement of Blowfish and CAST-128 algorithms and Security Analysis of Improved Blowfish Algorithm Using Avalanche Effect", IJCSNS, Vol.8 No.3, pp. 244-250, March 2008.