IJCTA, 9(10), 2016, pp. 4771-4777

Security Evaluation of Blowfish and Its Modified Version Using GT's One Shot Category of Nash Equilibrium

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

Abstract : In today's world, the importance and the value of transmitting data over the Internet or other media types are increasing. The need to protect the data from un- authorized users is increasing day by day. Providing security in a timely manner is one of the most challenging aspects in the internet and network applications. Cryptography can be defined as a tool to maintain confidentiality of information and to ensure its integrity and authenticity. There are two types of cryptography. One is the symmetric and the other is the asymmetric cryptosystem. Among symmetric cryptosystems, Blowfish is the best which is patent and licence free. As of today, the Blowfish has no cryptanalysis. An effort is made to enhance the security of the Blowfish cryptography algorithm by making modifications to the Feistel (F) function by combining the Blowfish and the Game Theory's one shot category in Nash Equilibrium (GTNE) concept. The outcome of the proposed GTNE-Blowfish and the existing Blowfish algorithm are analysed using Avalanche effect and the better performance of GTNE-Blowfish is reported.

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

1. INTRODUCTION

Network security refers to any activities designed to protect the network from malicious users. Cryptography is the art of science that protects the data from attacking forces and the unwanted actions of malicious users. Encryption is the process of transforming plain text data into cipher text in order to conceal its meaning and prevent any unauthorized user to retrieve the original data. 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, which in turn leads to more execution time and high energy consumption. Successful studies have been made to speed up the execution of cryptographic algorithms. There are a lot of benefits from parallel computing. The advantage of this system is its ability to handle large and extremely complex computations. The Blowfish algorithm was designed by Bruce Schneier to replace Data Encryption Standard, which was the Federal Information processing Standard Cryptography [1]. It is a symmetrical block cipher [2] having the advantages of secure, fast, easy to implement etc. For a cryptographic algorithm to be secure it should exhibit a strong Avalanche effect. Symmetric ciphers are fast and compact which requires less memory and power consumption, which is an important criteria for encryption techniques used in devices like PDAs and smart phones that have power, processor, and memory limitations. Parallel processing cannot be applied to the original Blowfish algorithm. So the structure of the Feistel function of the Blowfish algorithm is modified and a Game Theory's Nash Equilibrium (GTNE-Blowfish) algorithm is designed by combining Blowfish, Parallel Processing and Game Theory's

* 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

Optimization One-shot in Nash Equilibrium. These modifications give better performance in the security of the proposed algorithm thereby making the algorithm unbreakable. In one-shot category both the attacker and the defender are allowed to choose their strategies at the same time [3]. The basic idea of this research is to develop a simple, stronger and safer cryptographic algorithm.

2. RELATED WORK

A. Parallel processing

Parallel Processing Systems are designed to decrease the execution time of programs by splitting the program into multiple segments and processing them simultaneously. They are also referred to as multiprocessor systems or tightly coupled systems. In Parallel processing more than one computer processor is used to work on a problem at the same time. Coordinating the work of the individual processors in parallel processing makes the work little complicated when compared to working with a single processor. Sharing of resources and information by the processors must be done efficiently in parallel processing, as the work progresses. Parallel processors are used for problems which require a lot of computations with less time consumption. Parallel processing may be appropriate when the problem is very complicated to solve or when it is important to get fast results.

There are many possible ways of designing a parallel computer. One such way is categorizing them based on the two parameters, the stream of instructions that is, the algorithm and the stream of data that is, the input. The instructions can be carried out one at a time or concurrently, and the data can be processed one at a time or in multiples. Depending on the task to be performed, different parallel architectures have their own merits and demerits. A sequence of instructions for solving a problem that identifies the parts of the process that can be carried out simultaneously is a parallel algorithm. While writing programs for a parallel processor, the programmer must divide the problem into sub tasks and assign the sub tasks to appropriate processors. Then the programmer must decide the order in which the sub tasks are to be performed and when the communication to be sent to collect the results of the subtasks. There can be many algorithms for a particular problem, so the programmer needs to identify and implement the one best suited for a particular parallel architecture. According to Amdahl's Law the speed of an algorithm can be increased by parallel processing [4].

B. Game Theory

Game theory is the branch of mathematics involving the analysis of strategies for dealing with competitive situations where the outcome of a player's choice of action depends critically on the actions of the opponent player. Game theory deals with decision situations in which two intelligent players, the attacker and the defender have conflicting objectives [5]. Game Theory deals with playing strategies, so that winning (losing) is maximum (minimum) to each player. The objective of a game is to maximize the gain. The mathematical formulation of Game Theory is based on the Maxmin or Minmax criterion developed by J. Von Neumann [6]. Game theory is mainly used in economics, psychology, computer science, biology and political science.

In Game Theoretic settings players are assumed to be making choices that result in the most level of benefit or utility of individual. A lot of effort must be put to attain the nature of rational behavior, which in turn results in a long line of stability concepts. Cryptographic protocols are designed under the assumption that some parties are honest and stick to the protocol, while some parties are malicious and behave according to personal whim. In game theory's point of view, all parties are simply rational and behave in their own best interests. This viewpoint is incomparable to the cryptographic one, although no one can be trusted to stick to the protocol unless it is in their own best interests, the protocol need not prevent irrational behavior.

C. Nash Equilibrium

Nash equilibrium is a steady state of the play of a strategic game in which each player holds the correct expectation about the behaviour of other players and acts rationally. It does not attempt to examine the process by which a steady state is reached. Nash Equilibrium is a state in game theory where each player gets equal chance of winning the other. A Nash Equilibrium exists when there is no one side partiality move from any of the players

involved. In other words, no player in the game would be given a chance to take a different move as long as all the other players remain in the same strategy. Nash Equilibria are self-enforcing that is when players are at a Nash Equilibrium they do not wish to move because they will be in a more difficult position. Overall, an individual can receive no additional benefit from changing actions, assuming other players remain constant in their strategies. A game may or may not have Nash equilibria.

D. Materials And System Specifications

For this research the size of the text file ranges from 50 bytes to 208942 bytes. A Laptop with Intel Pentium T4500 @ 2.30 GHz CPU, 4.00GB Dual-Channel DDR3 and Linux Mint 17.1 is used. Avalanche effect, encryption time, decryption time, execution time, encryption throughput, decryption throughput and execution throughput are the performance metrics used for this research. The Blowfish cryptosystem is implemented using the C programming language in gcc compiler.

3. BLOWFISH ALGORITHM

Blowfish is one of the most common public domain encryption algorithms. Blowfish is a variable-length key block cipher. It is suitable for applications where the key does not change often. Blowfish Algorithm is a Feistel Network, which iterates an encryption function 16 times [7]. The block size is 64 bits, and the key can be any length from 32 bits to 448 bits [8]. The method invented by Horst Feistel to transform a function (F function) into a permutation is known as Feistel network. This is widely used in many block cipher designs, communication link, automatic file encryptor, etc.

The algorithm consists of two parts, a key expansion part and a data encryption part. A 64-bit plaintext message is first divided into 32 bits. XOR the left 32 bits with the first element of a P-array to create a value say P', run through the function F, then XOR with the right 32 bits of the message to produce a new value say F'. F' replaces the left half of the message and P' replaces the right half, and the process is repeated 15 more times with successive members of the P-array. XOR P' and F' with the last two entries in the P-array and recombine to produce the 64-bit cipher text. The Feistel structure of Blowfish algorithm is shown in Figure 1. The function F is obtained by dividing XL into four eight-bit quarters, *a*, *b*, *c*, and *d*.

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

The process of decryption is the same as encryption, but the key *p*-box is used in the reverse order. As Blowfish cipher is highly secure, fast, and suitable for different platforms, it has widespread application in the field of information security [9]. Blowfish is among the fastest block ciphers available. Blowfish is used in wide range of applications such as bulk encryption of data files, multimedia applications which 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]. A study was conducted for different popular key algorithms such as DES, 3DES, AES and Blowfish. They were implemented on two different platforms and their performance was compared by encrypting input files with different contents and sizes. The results showed that Blowfish was the best among all the other algorithms [11]. Bruce Schneier made a block cipher speed comparison among Blowfish, RC5, DES, IDEA, 3DES algorithms. The results showed the advantage of Blowfish among block ciphers in terms of speed. Since Blowfish algorithm is one of the fastest block ciphers and has no cryptanalysis, it was decided to enhance the security of Blowfish in this research paper.

4. PROPOSED GAME THEORY'S NASH EQUILIBRIUM BLOWFISH ALGORITHM AND ANALYSIS

A. GTNE-Blowfish

The structural diagram of the proposed GTNE-Blowfish algorithm which is obtained by combining Parallel processing, Blowfish and Game Theory's One-shot category of Nash Equilibrium is shown Figure 2.

A rational attacker attacks only the nodes in the sensible target set. The sensible target is a set of nodes whose security assets are the most wanted nodes to the attacker. The security asset refers to the confidentiality of data processed by the nodes. The objective of the attacker is to collect the maximum amount of information from the defender where the defender tries to protect the data [12]. For this research, investigation is done where both the attacker and the defender take the decisions at the same time by taking into account each other's strategy. This type of iterations falls under the one-shot game category [13]. In one-shot category the defender need not encrypt those nodes that are the least wanted by the attacker.

Parallel processing cannot be done in the original Blowfish algorithm. The proposed GTNE-Blowfish algorithm is a modified Blowfish algorithm. The modification is done in the F function. Game Theory's one-shot game category in Nash equilibrium state is incorporated in the blowfish algorithm and the F function is modified in such a way that parallel processing can be done. The modification shows parallel evaluation of different operations within the function. Without violating the security requirements, the Blowfish function F can be modified as follows

$$F(XL) = ((s4, d - S3, c) - ((D2, B * S4, d) + (S1, a * S4, d))) / (S4, d - (S2, b * S4, d))$$

The description of modification in the F function is as follows: In the first step a parallel evaluation of one subtraction operation and three multiplication operations, in the second step a parallel evaluation of one addition and one subtraction operation, in the third step a subtraction and in the fourth step a division operation.

B. Performance Comparisons

The performance of Blowfish algorithm and GTNE-Blowfish algorithm are compared using the performance metrics execution time, encryption time, decryption time and, avalanche effect. The encryption time, the decryption time, the execution time, is low for Blowfish algorithm than GTNE-Blowfish algorithm. But the Avalanche Effect is high for GTNE-Blowfish than Blowfish. GTNE-Blowfish is the best in terms of security. Blowfish algorithm by itself is highly secure. But above all GTNE-Blowfish is unbreakable in any circumstances.

5. EXPERIMENTAL RESULTS

According to Avalanche Effect, a change in one bit of the plain text or one bit of the key produces change in number of bits in the cipher text [14]. An encryption algorithm is said to be secure if a small change in either the plain text or the key produces a significant change in the cipher text. If the changes are small, this might provide a way to decrease the size of the plain text or key space to be searched and hence makes the encryption algorithm insecure. In this research, a change in the value of one bit in the input and the resulting avalanche effect for each algorithm is obtained for 20 times and the average value of the avalanche effect was taken and compared for the two algorithms Blowfish and GTNE-Blowfish. Blowfish had an average avalanche value of 57.1 and GTNE-Blowfish 66.7. Tabulation of results observed by changing one bit of plain text in the sample is shown in Table 1. Figure 3 represents the Security comparison of Avalanche effect of Blowfish algorithm and GTNE-Blowfish algorithm. In the bar chart Blowfish is represented as BF and GTNE-Blowfish as GTNEBF. The Blowfish algorithm has the lowest Avalanche effect when compared to the GTNE-Blowfish algorithm discussed here. So it is clear that GTNE-Blowfish algorithm is more secure than Blowfish algorithm.

Encryption Time is one of the performance metrics which is defined as the time required for converting plaintext message to cipher text at the time of encryption. Tabulation of results of encryption time with different packet size for Blowfish algorithm is shown in Table 2 and GTNE-Blowfish algorithm in Table 3. The encryption time of GTNE-Blowfish algorithm is slightly more than Blowfish algorithm.

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 different packet size for Blowfish algorithm and GTNE-Blowfish algorithm are shown in Table 2 and Table 3 respectively. The decryption time for GTNE-Blowfish algorithm is slightly more than Blowfish algorithm.

Execution time of an algorithm directly depends on the function of the algorithm and it clearly defines that more complex structure gives 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 different packet size for Blowfish algorithm is shown in Table 2 and GTNE-Blowfish algorithm in Table 3. The execution time taken by GTNE-Blowfish algorithm is slightly more than Blowfish algorithm which is reasonable.

The total plaintext in Megabytes divided by the Encryption time in seconds gives the Encryption Throughput.

Throughput = Total Plaintext in Mega Bytes / Encryption Time

Likewise, the total plaintext in Megabytes divided by the Decryption time in seconds gives the Decryption Throughput and the total plaintext in Megabytes divided by the Execution time in seconds gives the Execution Throughput.

The higher the value of throughput more is the efficiency of encrypting any text with an encryption algorithm. Tabulation of results of throughput with different packet size for Blowfish algorithm and GTNE-Blowfish algorithm are shown in Table 2 and Table 3 for Blowfish and GTNE-Blowfish algorithms respectively.

Table 1. Security Comparison - Avalanche Effect

Algorithm	*BF*	*GTNEBF*
Avalanche	57.1	66.7

Table 2. 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 3. Speed Analysis of GTNE-Blowfish Algorithm

Data size in Bytes	*Encryption*	*Decryption*	*Execution*
50	1.027	1.029	1.1677
60	1.0359	1.0381	1.1869
100	1.072	1.0738	1.2592

Data size in Bytes	*Encryption*	*Decryption*	*Execution*
250	1.2087	1.2107	1.5332
325	1.2772	1.2792	1.67
700	1.6219	1.7454	2.4881
900	1.8067	1.9431	2.8701
965	1.8631	2.0061	2.9911
5350	5.8342	4.173	9.1128
7400	7.0391	5.2289	11.2608
9000	8.0262	5.23482	12.3921
51202	25.3239	21.9479	46.3938
61442	29.7127	26.0667	54.9067
102402	46.9934	42.8102	88.9297
208942	85.9379	85.7631	171.3413
Average Time (millisec)	14.65199333	13.50333467	27.30023333
Throughput (MB/sec)	1.948693625	2.114458887	1.045860878

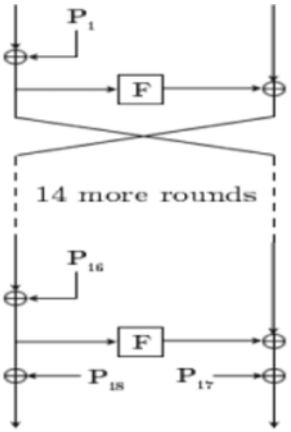

Fig. 1. Fiestel structure of Blowfish algorithm.

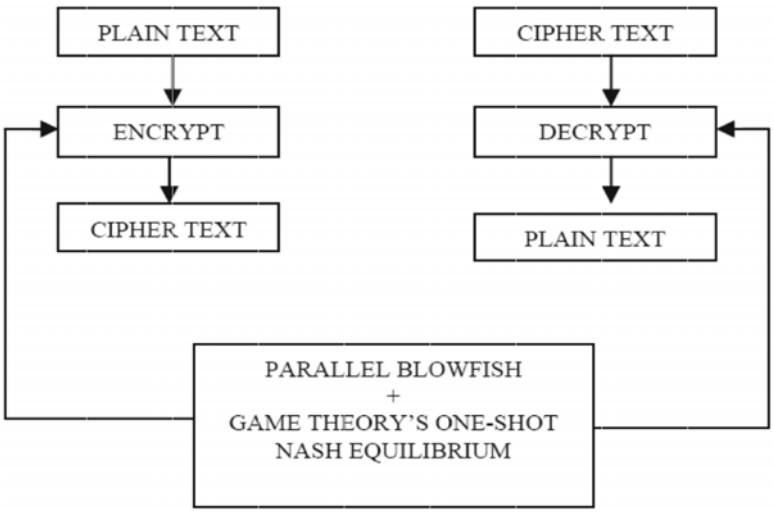

Fig. 2. Structural Diagram of GTNE-Blowfish Algorithm.

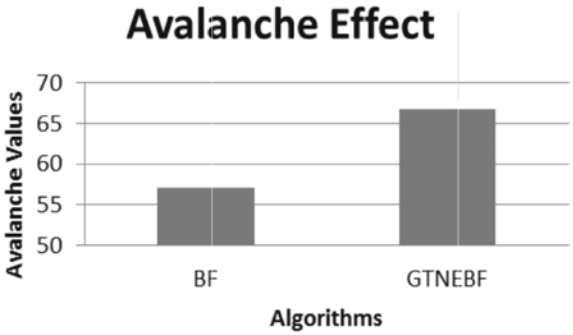

Fig. 3. Security Comparison - Avalanche Effect

6. CONCLUSIONS

This paper gives a detailed study of the most popular symmetric key encryption algorithm that is Blowfish and discussed about its advantages. Based on the benefits and weakness of the Blowfish algorithm, a new approach has been proposed and implemented to further enhance the existing algorithm to achieve better results in terms of security. The advantage of generating different cipher text for the same input is, it will give better performance in terms of the security aspect of the algorithm. Paper [15] says that an algorithm can be used for secure data transaction only if the encryption is strong. The above results clearly indicate that there is a huge variation in the Avalanche value for GTNE-Blowfish and Blowfish algorithm. It is clear that GTNE-Blowfish algorithm is very strong, secure and unbreakable than the Blowfish algorithm.

The research work can be further extended by doing the s-box calculation using normalization of random value to enhance the speed of the algorithm.

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. Ziad Ismail, Jean Leneutre, David Bateman, Lin Chen, "A Game Theoretical Analysis of Data Confidentiality Attacks on Smart-Grid AMI",IEEE Journal on Selected Areas in Communication", Vol.32, No.7, pp. 1486-1499, July 2014.
4. Roosta, Seyed H., " Parallel Processing and Parallel Algorithms: Theory and Computation", New York: Springer, 1999.
5. Hamdy A. Taha, "Operations research An Introduction", Sixth Edition, Prentice Hall of India, 1998.
6. C.B.Gupta, "Optimization Techniques in Operations Research", I.K.International Publishing House Pvt.Ltd., 2008.
7. Bruce Schneier, "The Blowfish Encryption Algorithm", *Dr. Dobb's Journal*, Vol. 19, No. 4, pp. 38-40, April 1994.
8. Bruce Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)", Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, pp. 191-204, 1994.
9. 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.
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. M.J. Osborne, Rubinstein, "A Course in Game Theory", Cambridge, MA, USA, MIT Press, 1994.
14. Krishnamurthy G.N., V.Ramaswamy, Leela G.H., Ashalatha, "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.
15. 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.