International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 20 (2017) pp. 9236-9244
© Research India Publications. http://www.ripublication.com

Performance Enhancement of Blowfish Encryption Using RK-Blowfish Technique

B. Shamina Ross
Assistant Professor, Department of Computer Applications,
Scott Christian College, Parvathipuram, Nagercoil,
Affiliated to Manonmaniam Sundaranar University, Tamilnadu, India.
Orcid Id: 0000-0002-9921-3406

V. Josephraj
Associate Professor and Head, Department of Computer Science,
Kamaraj College, Thiruchendur Road, Thoothukudi,
Affiliated to Manonmaniam Sundaranar University, Tamilnadu, India.

Abstract

Security is one of the biggest concerns in communications and electronic applications. In this paper, we have modified the Blowfish algorithm by enhancing its performance in terms of speed, Throughput, Power consumption and Avalanche effect. We have proposed a way to enhance the performance of the Blowfish cryptography algorithm by introducing parallel processing technique and making modifications to the Fiestel (F) function of Blowfish by combining the Blowfish and the Runge-kutta (RK) Method. The F function of Blowfish has been modified with different formulae and the outcome of a series of RK-Blowfish algorithms were compared with the Blowfish algorithm. The enhanced performances of RK-Blowfish series of algorithms are reported. This work would be useful to most consumer electronic appliances involved in data storage, transmission and communication ensuring data security.

Keywords: Avalanche Effect, Blowfish, Feistel Network, Parallel processing, Runge-Kutta, Throughput

INTRODUCTION

Cryptography has been in use for thousands of years. During this period, different forms of cryptosystems were developed and one of them is the Blowfish which is a fast, patent free symmetric block cipher and has no cryptanalysis. The Federal Information processing Standard Cryptography [1], Data Encryption Standard [DES] was replaced by the new algorithm Blowfish designed by Bruce Schneier. Blowfish is the best symmetrical block Cipher [2] having the advantage of easy implementation, being secure and fast. The Blowfish algorithm uses only two operations XOR and addition on 32-bit words. Blowfish uses only 4KB or even a lesser memory when it runs. However, one limitation is that, Parallel processing technique which has the advantage to handle large and extremely complex computations cannot be applied to Blowfish algorithm. Hence, the objective of this study is to analyze the Blowfish algorithm and improve its performance using Parallel Processing and RK technique thereby simplifying complicated cryptographic algorithms by splitting up their tasks to run simultaneously and to increase the execution speed and lower the power consumption. RK methods are self- starting and easy to program for digital computers.

The speed of the algorithm can be enhanced by parallel processing [3]. In the present study, Parallel Processing, Blowfish and Runge-Kutta method are combined with the objective to improve the performance. Blowfish has one of the most sought-after diffusions namely a potential security [4]. The Avalanche effect has been used to show that the proposed series of RK-Blowfish algorithms possess potential diffusion characteristics such as security as that of the original Blowfish algorithm. Thus the proposed RK-Blowfish algorithms enhance the performance over Blowfish.

As RK-Blowfish is a block cipher with a varying key size, it is most suitable for automatic file encryption or communications link. RK-Blowfish algorithm can be widely used in Electronic communications, E-Commerce software, E-Mail encryption, On-line chat and Password Management.

RUNGE-KUTTA METHOD

Runge-Kutta methods are well-suited for practical applications when compared to Taylor Series and Euler Method. Taylor series requires evaluation of partial derivatives of higher orders manually which is not possible in any practical application. Therefore, methods which do not require evaluation and computation of higher order derivatives are needed. The most important class of methods in this direction is the Runge-Kutta methods. The truncation error for Taylor series is $O(h^4)$, Euler Method is $O(h^2)$ and the Runge-kutta method is $O(h^5)$ [5]. By using large values of h, the good stability properties of Runge-Kutta methods can be utilized [6]. The Euler's method is neither very accurate nor very stable, so it is not suitable for practical purposes. Accuracy is the best in the Runge-Kutta methods [7] and the error value is low. It is the most accurate of numerical approximation techniques. Runge-Kutta methods belong to a family of one-step method. In one step method the global error is of the same order as local error. In one-step methods, the information from only one preceding point is considered, that is to estimate the value y_i it needs the conditions at the previous point y_{i-1} only. They are all based on the general form of the extrapolation equation,

$$\begin{aligned} y_{i+1} &= y_i + slope\ x\ interval\ size \\ &= y_i + mh \end{aligned}$$

where m represents the slope that is weighted averages of slopes at various points in the interval h.

Runge-Kutta method can be easily incorporated into relatively simple driver schemes as it treats each step in a sequence of steps identically. Preceding behavior of a step is not considered in its succeeding step. This is mathematically justified by the statement that any point along the trajectory of an ordinary differential equation can serve as an initial point. Runge-Kutta methods are easy for Automatic Error Control.

An RK method is called the r-order Runge-Kutta method when slopes at r points are used to construct the weighted average slope m. Since Euler's method uses only one slope at (x_i,y_i) to estimate y_{i+1}, it is a first-order Runge-Kutta method. As Heun's method employs slopes at two end points of the interval, it is a second-order Runge-Kutta method. The higher the order better would be the accuracy of estimates [8]. RK methods are self-starting and easy to program for digital computers. Second order Runge-Kutta methods are obtained using two slopes in the Runge-Kutta methods. The method has one arbitrary parameter, whose value is suitably chosen. The methods using four evaluations of slopes have two arbitrary parameters. The values of these parameters are chosen such that the method becomes simple for computations. The following method is one such choice

$$\begin{aligned} y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), where k_1 \\ = hf(x_i, y_i), k_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right), k_3 \\ = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right), k_4 \\ = hf(x_i + h, y_i + k_3) \end{aligned}$$

This is called the classical Runge-Kutta method of fourth order or simply fourth order Runge-Kutta method. When five slopes are used, we do not get a fifth order method. So the Runge-Kutta fourth order method is preferred for computations [9].

The second order method requires two evaluations of f at every time step. In general, for an rth order Runge-Kutta method S(r) evaluations of f for each time step, where

$$S(r) = \begin{cases} r & for\ r \le 4 \\ r+1 & for\ r = 5\ and\ r = 6 \\ \ge r+2 & for\ r \ge 7 \end{cases}$$

An appropriate compromise of the computing requirements of a low truncation error per step and a low computational cost per step is represented by the fourth-order Runge-Kutta integration. The error convergence is the best for Runge-Kutta method. The 3/8 method is a version of fourth order Runge-Kutta method for approximating the solution of the initial value problem $y'(x) = f(x,y);\ y(x_0) = y_0$ which evaluates the integrand, $f(x,y)$, four times per step. For step $i+1$,

$$y_{i+1} = y_i + \frac{1}{8}(k_1 + 3k_2 + 3k_3 + k_4), where$$

$$k_1 = hf(x_i, y_i)$$

$$k_2 = hf\left(x_i + \frac{h}{3}, y_i + \frac{k_1}{3}\right)$$

$$k3 = hf\left(x_i + 2\frac{h}{3}, y_i - \frac{k_1}{3} + k_2\right)$$

$$k4 = hf(x_i + h, y_i + k_1 + k_2 + k_3)\ and$$

$$x_i = x_{0+ih}$$

The fourth order Runge-Kutta method with modification needs less storage requirements than the other Runge-Kutta formulae. If the exact solution of the difference equation p without round-off error tends to the solution of the partial differential equation P, as Δx and Δt both tend to zero then the numerical method is said to be convergent. The difference (P - p) is termed as the discretization error. For linear problems if the conditions of stability and compatibility are satisfied by the numerical method under consideration, then the scheme is said to be convergent. For RK-Method in order to test the convergence, the control of accuracy and the adjustment of step size Δt is done by comparison of the results due to double and single step size $2\Delta t$ and Δt. The RK approach possesses the advantage of higher order accuracy and automatic step size, Δt [10]. High order methods are capable of achieving highly accurate approximations of differential equations solutions at lower computational cost than low order methods [11].

Blowfish Algorithm

The Blowfish algorithm takes a 64-bit plaintext as input and then gives an output of 64-bit cipher text. The key length varies from 32 bits to 448 bits [12]. The algorithm consists of two main actions. One is the key expansion and the other is the data encryption. During key expansion, a key of maximum size of 448 bits is converted into several subkey arrays to a total of 4168 bytes. The original subkey p-box and s-box are fixed. They are initialized in order with a fixed string that consists of hexadecimal digits of Pi minus the initial value 3. After key expansion, the data encryption takes place in a 16-round Feistel network [13]. A key dependent permutation and a key and data dependent substitution take place in each round. A key p-box [18], a key s-box [4] [256], and a core Feistel function are used by the algorithm. The subkeys are computed ahead of 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 mod2^32) XOR S3,c) + S4,d mod2^32

Here, "+" is addition on 32-bit words, and XOR is 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. The Feistel structure of Blowfish algorithm is shown in Figure 1.

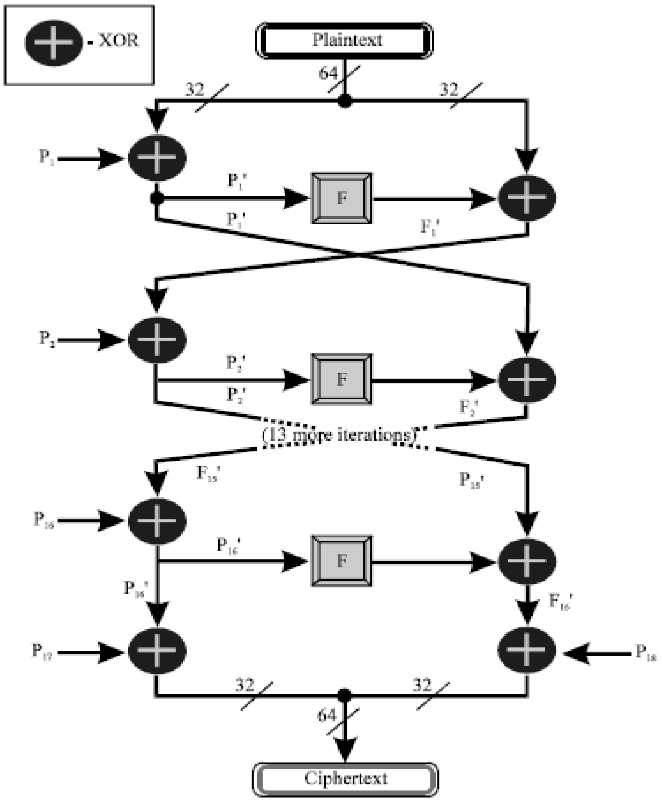

Figure 1: Feistel Structure of Blowfish Algorithm

The principle of Blowfish algorithm is both easy to understand and implement. All subkeys of Blowfish are influenced by every bit of the key. The key and the data get mingled together completely, making it difficult to analyze the key. A great Avalanche effect is produced by the function F.

Blowfish is among the fastest block ciphers currently available. Blowfish can be used for bulk encryption of data files, encryption of voice and media files in multimedia and remote backup of hard disk. Geographical information system uses blowfish for cryptographic protection of sensitive data. These applications are run at high-end servers and workstations. Moreover, they are also run where there is a demand for high speed encryption and higher throughput and a bulk of data is processed [14]. A performance comparison of various algorithms such as DES, 3DES, AES and Blowfish was conducted [15] by encrypting input files of varying contents and sizes on two different hardware platforms. The results showed that Blowfish was more efficient than that of the other algorithms. Bruce Schneier compared the block ciphers Blowfish, RC5, DES, IDEA and 3DES in terms of speed. The results showed the advantage of Blowfish over block ciphers in terms of speed and also it was clear that, the security of the Blowfish algorithm is very promising. As Blowfish algorithm is secure, fast and suitable for different platforms, it is widely used in the field of information security [16].

Proposed RK-Blowfish Algorithms And Analysis

The block diagram of the proposed RK-Blowfish algorithm which is obtained by combining Parallel processing, Blowfish and Runge-Kutta technique is shown in Figure 2. The series of RK-Blowfish algorithms is detailed below.

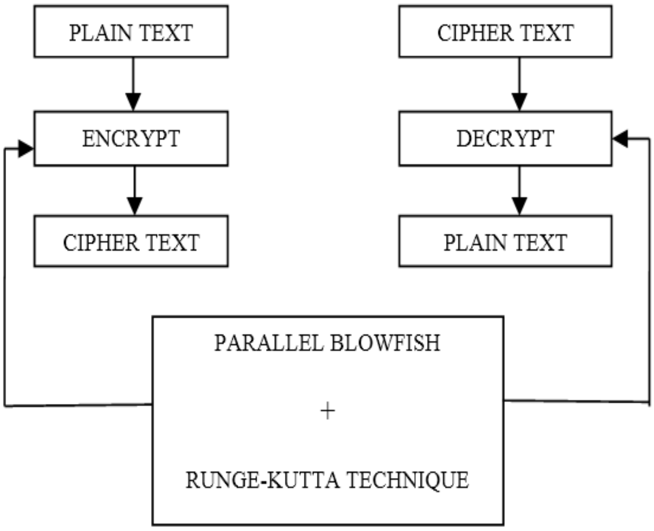

Figure 2: Block Diagram of RK-Blowfish Algorithm

RK-Blowfish1

The proposed RK-Blowfish1 algorithm is obtained with a modification in the F function of Blowfish algorithm by incorporating the parallel processing and Runge-Kutta technique in the F function of the Blowfish algorithm. The modification shows parallel evaluation of different operations within the function. Without violating the security requirements and also to increase the speed, the modification in the Blowfish function F can be incorporated by parallel evaluation of different operations such that

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

This modification supports the parallel evaluation of two addition operations (S1, a + S2, b mod 2^32) and (S3, c + S4, d mod 2^32) using threads and two shift operations, also parallel evaluation of two addition operations and finally an addition operation. The parallel evaluation reduces the time consumed for two additions to one and the shift operation also executes simultaneously.

RK-Blowfish2

The F function is modified so that the speed and security metrics are increased.

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

This modification supports parallel evaluation of two addition operations (*S1, a + S2, b mod 2^32*) and (*S3, c + S4, d mod 2^32*) using threads and two shift operations, also parallel evaluation of two addition operations and finally an XOR operation. The parallel evaluation reduces the time consumed for two additions to one and the shift operation also takes place simultaneously. Moreover, the parallel evaluation of two addition operations is reduced to the time of one addition operation.

RK-Blowfish3

The F function is modified in such a way that the efficiency of RK-Blowfish3 is enhanced than that of the Blowfish algorithm in terms of security.

$$F(XL) = ((S1,a\ XOR\ S2,b) + (S3c,XOR\ S4,d) \bmod 2^{32}) \\ + (<<<S2,b\ XOR\ <<<S3,c) \bmod 2^{32}$$

This modification supports the parallel evaluation of two XOR operations (*S1, a XOR S2*) and (*S3, c XOR S4, d*) using threads and two shift operations, also parallel evaluation of one addition operation and one XOR operation and finally an addition operation. The parallel evaluation reduces the time from two XOR operations to time consumed for one XOR operation and two shift operations also take simultaneosly. Then the parallel evaluation of one addition operation and one XOR operation is reduced to the time consumed for one addition operation.

RK-Blowfish4

The F function is modified in such a way that the efficiency of RK-Blowfish4 is enhanced than that of the Blowfish algorithm in terms of security.

$$F\ (XL) = (S1,a\ XOR\ S2,b)\ XOR\ (S3,c\ XOR\ S4,d)\ XOR \\ (<<<S2,b\ XOR\ <<<\ S3,c)$$

This modification supports the parallel evaluation of two XOR operations (*S1, a XOR S2*) and (*S3, c XOR S4, d*) using threads and two shift operations, also parallel evaluation of two XOR operations and finally one XOR operation. The parallel evaluation of two XOR operations and two shift operations is reduced to the time consumed for one XOR operation, also the parallel evaluation of two XOR operations to the time of one XOR operation and finally one XOR operation.

In the above four methods, all the operations take place in 3 steps. As the algorithm uses 16 iterations, the time is reduced 16 times for every encryption and decryption. This is a significant enhancement.

Performance Comparisons

In this paper the performance metrics namely Execution time, Encryption time, Decryption time, Throughput, Avalanche effect and Power consumption are used to evaluate the Blowfish, RK-Blowfish1, RK-Blowfish2, RK-Blowfish3 and RK-Blowfish4 algorithms. The encryption time is the least for RK-Blowfish3 followed by Blowfish, RK-Blowfish1, RK-Blowfish2 and RK-Blowfish4. The decryption time is the least for RK-Blowfish1 followed by RK-Blowfish2, Blowfish, RK-Blowfish3 and RK-Blowfish4. The Execution time is the least for RK-Blowfish1 followed by RK-Blowfish2, Blowfish, RK-Blowfish3 and RK-Blowfish4. The Throughput is the highest for RK-Blowfish1 followed by RK-Blowfish2, Blowfish, RK-Blowfish3 and RK-Blowfish4. Higher the value of Throughput, greater is the efficiency of encrypting any text with an encryption algorithm. If the throughput is high the power consumption is low [17]. The power consumption is the least for RK-Blowfish1 followed by, RK-Blowfish2, Blowfish, RK-Blowfish3 and RK-Blowfish4. The Avalanche Effect is the highest for RK-Blowfish4 followed by RK-Blowfish1, RK-Blowfish2, RK-Blowfish3 and Blowfish. Of all the algorithms discussed, RK-Blowfish1 is the best in terms of speed and security.

EXPERIMENTAL RESULTS

For this research the experimentation was done with file size varying from 50 bytes to 208942 bytes. For each file size the experiment was repeated for twenty times and the average of the twenty values was taken. The system was used continuously without any break in between and no hang-up of the system was experienced. There was a slight variation in the timings because of the system process time that is, the CPU clock cycle. The size of the plain text and the cipher text was the same in all the cases. A Laptop with Intel Pentium T4500 @ 2.30GHz CPU, 4.00GB Dual-Channel DDR3 and Linux Mint 17.1 was used in which the performance data were collected. The performance metrics were the encryption speed, decryption speed, execution time, encryption throughput, decryption throughput, and execution throughput, power consumption and avalanche effect. The RK-Blowfish algorithm is implemented using the C programming language in gcc compiler.

Encryption Time

During encryption, the time taken to convert plaintext message to cipher text is defined as the encryption time. Figure 3 shows the average encryption time for different input size for the encryption time with the five algorithms discussed here. In the bar chart, Blowfish is represented as BF, RK-Blowfish1 as RKBF1, RK-Blowfish2 as RKBF2, RK-Blowfish3 as RKBF3 and RK-Blowfish4 as RKBF4. It is clear from the bar chart that the average encryption time for RK-Blowfish3 algorithm is the least among the five algorithms compared here. Tabulation of results of encryption time for different packet size is shown in Table 1.

Table 1: Comparative Encryption Times (in milliseconds) Of Encryption Algorithms With Different Packet Size

Input Size in Bytes	BF	RKBF1	RKBF2	RKBF3	RKBF4
50	0.7586	0.7932	0.7725	0.7746	0.7772
60	0.7709	0.8001	0.7824	0.7821	0.7849
100	0.7919	0.8284	0.8061	0.8095	0.8151
250	0.8962	0.9373	0.911	0.9103	0.9181
325	0.9486	0.9859	0.9602	0.9588	0.9714
700	1.2776	1.2576	1.2224	1.2198	1.2321
900	1.3354	1.3954	1.3564	1.3602	1.3699
965	1.3741	1.4445	1.4041	1.4042	1.4139
5350	4.5246	4.5769	4.6376	4.5666	4.5843
7400	5.9181	6.1462	5.9795	5.9823	5.985
9000	6.9128	7.1543	6.8138	6.8542	6.9218
51202	20.9473	26.5703	19.3078	20.2595	22.4491
61442	23.8123	24.4331	28.525	23.726	23.1613
102402	37.7555	26.805	33.9861	35.4657	35.8265
208942	63.2736	67.3147	64.4987	64.5982	65.607
Average Time (msec)	11.41983333	11.42952667	11.46424	11.31146667	11.52117333
Throughput (Mb/sec)	2.500233162	2.498112724	2.490548523	2.524186017	2.475655837

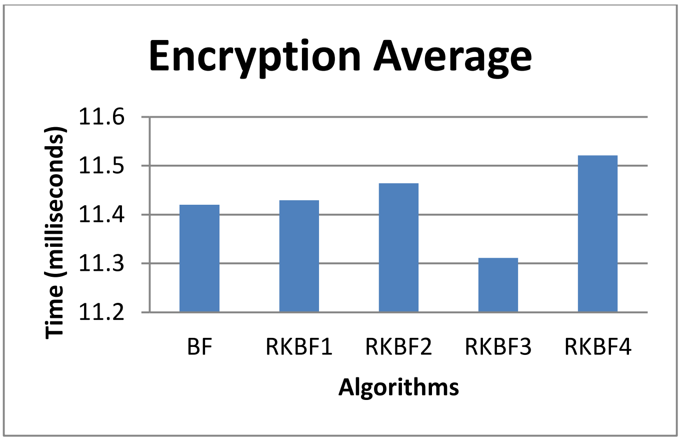

Figure 3: Comparison of Average Encryption Time

Decryption Time

During decryption, the time taken to convert plaintext message to cipher text is defined as the encryption time. Figure 4 shows the average decryption time for different input size for the encryption time with the five algorithms discussed here. It is clear from the bar chart that the amount of decryption time taken by RK-Blowfish1 algorithm is the least among the five algorithms compared here. Tabulation of results of decryption time for different packet size is shown in Table 2.

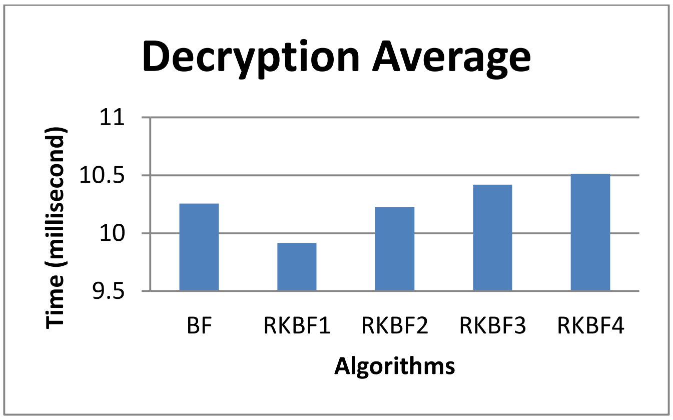

Figure 4: Comparison of Average Decryption Time

Table 2: Comparative Decryption Times (in milliseconds) Of Decryption Algorithms With Different Packet Size

Input Size in Bytes	BF	RKBF1	RKBF2	RKBF3	RKBF4
50	0.7602	0.7948	0.7743	0.7762	0.779
60	0.7722	0.802	0.7839	0.7835	0.7871
100	0.7934	0.8303	0.8075	0.8107	0.8166
250	0.8978	0.9391	0.913	0.9115	0.9197
325	0.9497	0.9879	0.9616	0.9605	0.973
700	1.2005	1.2595	1.2257	1.2213	1.2341
900	1.3364	1.4969	1.3585	1.3618	1.3992
965	1.549	1.5374	1.6198	1.4057	1.4548
5350	4.4654	4.5679	3.6188	4.2647	4.2736
7400	5.8499	5.223	3.7039	5.3527	4.3659
9000	5.1447	5.2213	4.6715	5.0985	5.8206
51202	16.2216	17.389	16.5845	16.8258	17.3665
61442	19.2313	20.2018	19.5559	19.6535	19.878
102402	31.5148	20.1584	32.2707	32.2367	32.5358
208942	63.159	67.3096	64.5454	64.6135	65.0877
Average Time (msec)	10.25639333	9.914593333	10.22633333	10.41844	10.51277333
Througput (Mb/Sec)	2.783848579	2.879820184	2.792031619	2.740549065	2.715957541

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. Figure 5 shows the average execution time for different input size for the five algorithms compared here. It is clear from the bar chart that the execution time for RK-Blowfish1 algorithm is the least among the five algorithms compared here. Tabulation of results of execution time with different packet size is shown in Table 3.

Table 3: Comparative Execution Times (in milliseconds) With Different Packet Size

Input Size in Bytes	BF	RKBF1	RKBF2	RKBF3	RKBF4
50	0.8875	0.9229	0.9009	0.9042	0.9075
60	0.9058	0.9374	0.9185	0.9184	0.9228
100	0.9543	0.9936	0.9684	0.9725	0.9558
250	1.1592	1.2098	1.1779	1.1766	1.1865
325	1.2615	1.3105	1.2771	1.2764	1.2915
700	1.7646	1.8475	1.8	1.7953	1.8135
900	2.0352	2.2428	2.0698	2.0849	2.1206
965	2.29	2.3211	2.388	2.1617	2.2044
5350	8.3683	8.4852	7.6134	8.1986	8.2187
7400	11.146	10.7002	9.0238	10.6954	9.6924
9000	11.4318	11.7041	10.8329	11.3054	12.0928
51202	36.5376	43.2946	34.9996	36.4537	39.1718
61442	42.4173	43.9768	47.4846	42.7473	42.3971
102402	68.651	46.7442	65.6169	67.0669	67.7206
208942	126.085	134.2725	128.821	128.9573	130.4314
Average Time (msec)	21.05967333	20.73088	21.05952	21.11430667	21.40849333
Throughput (Mb/sec)	1.3557782	1.377280945	1.355788071	1.352270119	1.333687783

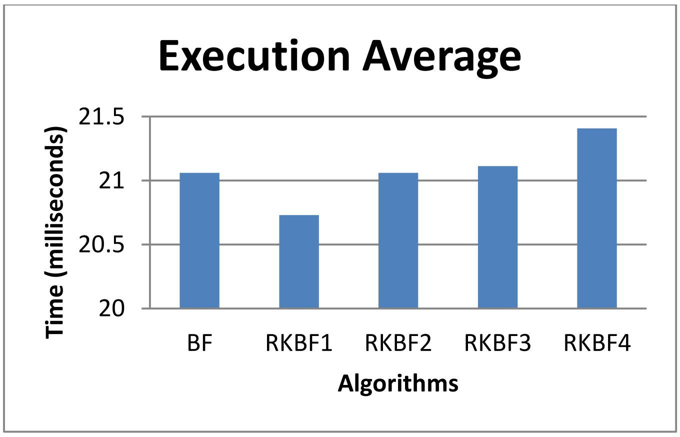

Figure 5: Comparison of Average Execution Time

Throughput

The throughput of the encryption scheme is calculated by dividing the total plaintext in Megabytes encrypted on the total encryption time 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.

Figure 6 shows the comparison of Encryption Throughput of Blowfish, RK-Blowfish1, RK-Blowfish2, RK-Blowfish3 and RK-Blowfish4 algorithms with different input packet sizes. The bar chart clearly shows that the RK-Blowfish3 algorithm has the highest Encryption Throughput among the five algorithms discussed here. The Encryption Throughput values are shown in Table I.

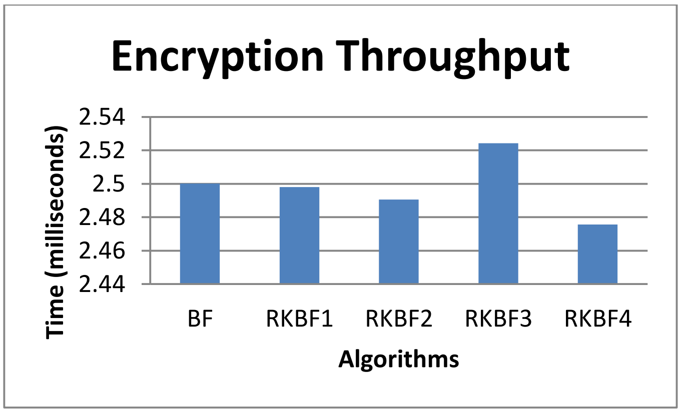

Figure 6: Comparison of Encryption Throughput

Figure 7 shows the comparison of Decryption Throughput of Blowfish, RK-Blowfish1, RK-Blowfish2, RK-Blowfish3 and RK-Blowfish4 algorithms with different input packet sizes. From the graph it is clear that the RK-Blowfish1 algorithm has the highest Decryption Throughput among the five algorithms discussed here. The Decryption Throughput values are shown in Table 2.

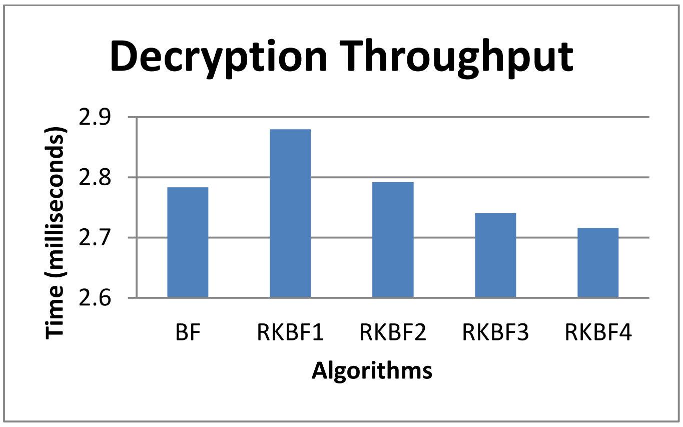

Figure 7: Comparison of Decryption Throughput

Figure 8 shows a bar chart representing the comparison of Execution Throughput of Blowfish, RK-Blowfish1, RK-Blowfish2, RK-Blowfish3 and RK-Blowfish4 algorithms with different input packet sizes. From the bar chart it is clear that the RK-Blowfish1 algorithm has the highest Execution Throughput among the five algorithms compared here. The Execution Throughput values are shown in Table 3.

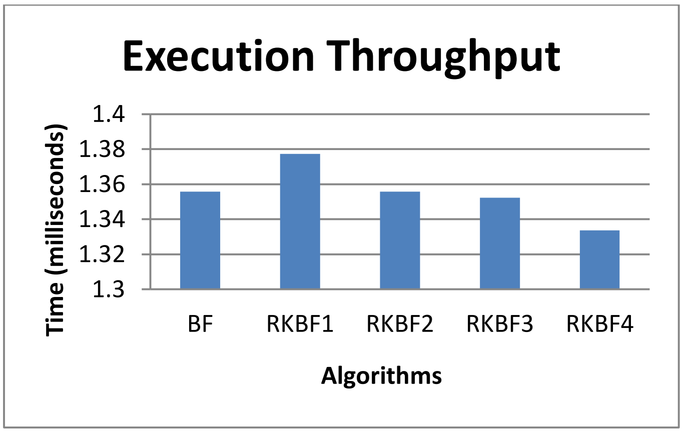

Figure 8: Comparison of Execution Throughput

The higher the Throughput less will be the power consumption. So from the above findings it is clear that the power consumption will be the least for RK-Blowfish1 algorithm which has the highest Execution Throughput among the five algorithms compared here.

Avalanche Effect

When a change in one bit of the plain text or one bit of the key schedule produces a change in many bits of the cipher text, it is called Avalanche Effect [18]. A desirable feature of any encryption algorithm is that a small change in either the plain text or the key should produce a significant change in the cipher text. If the changes are small, this might provide a way to reduce the size of the plain text or key space to be searched and hence makes the cryptanalysis very easy. For a cryptographic algorithm to be secure it should exhibit strong Avalanche effect. Thus higher the Avalanche value, higher will be the security. Figure 9 is a bar chart that represents the Avalanche effect of Blowfish algorithm, RK-Blowfish1 algorithm, RK-Blowfish2 algorithm, RK-Blowfish3 algorithm and RK-Blowfish4 algorithm.

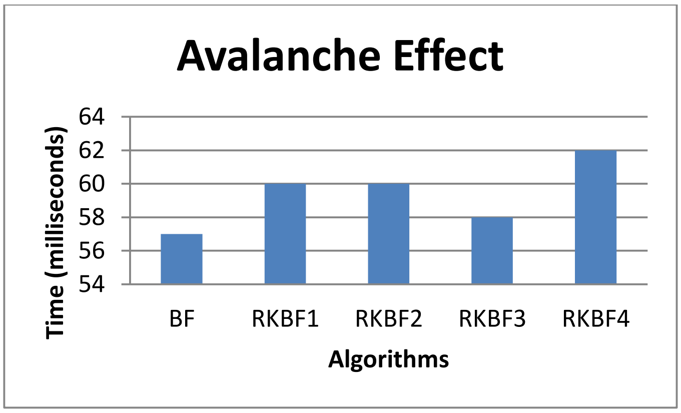

Figure 9: Avalanche Effect Comparison

The Blowfish algorithm has the lowest Avalanche effect when compared to the RK-Blowfish series algorithms discussed here. So it is clear that RK-Blowfish4 algorithm, RK-Blowfish1 algorithm, RK-Blowfish2 algorithm and RK-Blowfish3 algorithm are more secure than Blowfish algorithm. Tabulation of results observed by changing one bit of plain text in the sample is shown in Table 4.

Table 4: Avalanche Effect Comparison

Encryption Technique	Avalanche Effect
BF	57
RKBF1	60
RKBF2	60
RKBF3	58
RKBF4	62

CONCLUSION

Based on the results of the studies made, the proposed algorithm has the following advantages over the existing Blowfish algorithm. The first advantage is, different cipher text is generated for the same input which greatly enhances the security aspect. This is because a new random number gets generated each time and this, as a result gives difference in the application of Feistel (F) function over each round. The second great advantage of this approach is that, it is less time consuming as compared to that of Blowfish algorithm since the Parallel processing technique is applied. The third advantage is the Throughput which is higher than that of the existing Blowfish. The fourth advantage is the high security metric which is the result of high Avalanche value. The above results clearly indicate that the Encryption time, Decryption time, Throughput, Avalanche effect and Power consumption of RK-Blowfish1, RK-Blowfish2, RK-Blowfish3, RK-Blowfish4 are much more efficient than Blowfish algorithm. RK-Blowfish can be used in consumer electronic devices such as PDAs and smart phones which require less memory and power consumption. It can also be used in personal database programs, encryption in removable media, clinical data collection, biometric identification and authentication, using voice, facial or finger print recognition. This study can be further extended with optimization techniques which have high potentials.

REFERENCES

[1] U.S. National Bureau of Standards, “Data encryption standard”, U.S. Fed.Inform. Processing Standards Pub., FIPS PUB 46, January 1977, pp. 2-27.

[2] Bruce Schneier, “Applied Cryptography: Protocols, Algorithms, and Source Code in C”, 2nd Edition, New York, John Wiley and Sons, Inc. 1996, pp. 21-27.

[3] Gene M. Adahl, “Validity of the single processor approach to large scale computing capabilities”, in proceedings of the spring joint computer conference AFIPS’67(spring) ACM, Newyork, NY, USA, April 18-20, 1967 pp. 483-485.

[4] William Stallings, “Cryptography and Network Security”, Fifth Edition, Pearson Education, 2011, pp. 119-120.

[5] M.K.Jain, S.R.K. Iyengar, R.K.Jain, “Numerical Methods for Scientific and Engineering Computation”, Fifth Edition, New Age International Publishers, 2007, pp. 438-445.

[6] L.F. Shampine, H.A.Watts, “Comparing Error Estimators for Runge-Kutta Methods”, Mathematics of Computation, Vol. 25, Number 115, July 1971, pp.445-455.

[7] S.S.Sastry, “Introductory Methods of Numerical Analysis”, Fourth Edition, 2009, pp. 304-306.

[8] E. Balagurusamy, “Numerical Methods”, Tata McGraw-Hill Education Private Limited, pp. 436-437.

[9] S.R.K.Iyengar, R.K.Jain, “Numerical Methods”, First Edition, New Age International Publishers, 2009, pp. 200-203.

[10] Ashok Kumar, T. E.Unny, “Application of Runge-Kutta method for the solution of non-linear partial differential equations”, Applied Mathematical Modelling, Elsevier, Vol.1, Issue4, March1977, pp. 199-204.

[11] J.C. Butcher, “A History of Runge-Kutta Methods”, Elsevier, Applied Numerical Mathematics, Vol. 20, 1996, pp. 247-260.

[12] 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.

[13] Bruce Schneier, “The Blowfish Encryption Algorithm,” Dr. Dobb's Journal, Vol. 19, No. 4, April 1994, pp. 38-40.

[14] 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 2005, pp. 625-639.

[15] Aamer Nadeem, Dr.M.Younus Javed, “A Performance Comparison of Data Encryption Algorithms”, IEEE, Information and Communication Technologies, ICICT 2005.First International Conference, 2006-02-27, 2005, pp. 84-89.

[16] 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, 2009, pp. 24-28.

[17] D.S.Abdul.Elminaam, H.M.Abdul Kader, M.M.Hadhoud, “Performance Evaluation of Symmetric Encryption Algorithms”, Communications of the IBIMA, ISSN: 1943-7765Vol.8, 2009, pp. 58-64.

[18] 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, March 2008, pp. 244-250.