Design and Development of a Rapid AES based Encryption Framework

Jasmeet Singh[1]
[1] M.tech Student,
Department of Computer Engineering,
Punjabi University, Patiala, Punjab, India

Harmandeep Singh[2]
[2]Assistant Professor,
Department of Computer Engineering,
Punjabi University, Patiala, Punjab, India

***Abstract:* - AES algorithm for encryption is a popular option for the developers and researchers. The AES algorithm is used to encrypt almost all types of data. The reason behind its wide adaptation is the robustness and unbreakable security levels provided or created by the Advanced Encryption Standard (AES) cryptography algorithm. It is a popularly saying that breaking into the simplest AES encryption may take years to break into. But the major setback arises when it comes to the encryption/decryption speeds of the AES algorithm. The slower speeds hinder the developers from using AES in some application where a smaller delay can cause various types of performance lags in such applications. The developers use other encryption algorithms for the security of data which adds less performance lag and are quicker than AES. But the use of other cryptographic algorithms may cause a major setback to security of such applications. The encryption algorithms like Blowfish are prone to several types of attacks and can be broken into easily when compared to the AES. Hence it becomes very important to improve the encryption/decryption speeds of the AES algorithm. In this paper, we have addressed the issue of speed of AES. The encryption and decryption speeds of AES has been improved by using various methods like programming optimization, effective data segmentation and aggregation method and static S-Box. The results have proved the effectiveness of the improved AES on image data. The Implemented algorithm is faster than the traditional AES.**

Keywords: AES, AES speed, Programming optimization, data validation algorithm, encryption speed, decryption speed

INTRODUCTION

Cryptography [1] is a technique by which secure communication can be done in the presence of third party [2]. It is an art and science of secret writing [3]. Data that can be read and understood without any special measures is called Plain Text. Data that cannot be easily read and understood is known as cipher text. The process by which Plain text is converted into a cipher text that cannot be easily understood by any unauthorized person is known as Encryption. The reverse process of the encryption, which converts back the cipher text to plaintext, so it can be easily understood, is known as Decryption. For encrypt and decrypt, there is need of key. A key is number or set of number that the cipher, as an algorithm, operates on.

Figure1: Encryption and Decryption [4]

Cryptography is of two Types'. First one is Symmetric key and second is Asymmetric key. In Symmetric Key Cryptography, the same key is used by both Sender and Receiver for the Encryption and Decryption. Symmetric [5] key algorithm is also known as Secret key and single key Encryption. AES, DES and Triple DES [6] are the Example of Symmetric Key Cryptography. Asymmetric key is also known as public key encryption this method of encrypting messages makes use of two keys: a public key and a private key. The public key is made publicly available and is used to encrypt messages by anyone who wishes to send a message to the person that the key belongs to. The private key is kept secret and is used to decrypt received messages. An example of asymmetric key encryption system is RSA.

Advanced Encryption Standard

The Advanced Encryption Standard [7] is a winner of contest, which is organized by the U.S Government in 1997. The AES was designed because DES was found too weak due to its small key size and Technological advancement in Processor power. Although 3DES increased the key size but it was too slow. The National Institute of Standards and Technology (NIST) [8] chose the Rijndael Algorithm, named after its two Belgian inventors, Joan Daemen and Vincent Rijmen as the basis of AES.AES is a very complex round cipher.AES works on fixed length group of bits, known as blocks [9]. It takes as a 128bit input and produces same size of output.AES uses three different key sizes: 128, 192 and 256 bits. While Rijndael supports variable key and block size in any multiple of 32, with minimum of 128 bit and maximum of 256 bit.

a_1	a_5	a_9	a_{13}
a_2	a_6	a_{10}	a_{14}
a_3	a_7	a_{11}	a_{15}
a_4	a_8	a_{12}	a_{16}

Figure 2: Example of 128-bit State of AES [10]

Number of rounds	BL = 128	BL = 192	BL = 256
KL = 128	10	12	14
KL = 192	12	12	14
KL = 256	14	14	14

Table1: Number of Rounds [11]

IMPLEMENTATION OF THE ALGORITHM

The algorithm has been designed with three major phases of development. The first phase of development associates with including the implementation of improved AES for changed window size or block size. The objective of second phase has been achieved by the improvement in the key-matrix for key-expansion and by improving the s-box size and shape for the effective and fit AES scheme. The last phase of the development is associated with the development of data validation and segmentation algorithm to make the encryption application fit for wider number of situations.

Phase 1: Programming Optimization

In this model, an improved version of AES encryption is been designed to achieve the goal of Rapid implementation of AES algorithm for software systems. In this Rapid AES system, in order to make the whole system run in a faster speed, several specific methods have been used in the data pass processing. Firstly, the input digits have been increased to 128 bits. This method will improve the operating speed of the whole system. 128 bits are set up at the input terminal together, so in one time sequence, all data will enter into the encryption or decryption system. This will reduce the data entering and passing time significantly.

Phase 2: Static S-Box

Secondly, in this design, after receiving the key matrix, every part of the Key-Expansion is under a continuously working state. Without waiting (one clock cycle for S-BOX), no performance of enhancement of the system can achieve. The Key-Expansion part is divided into two parts. One part takes the responsibility for calculating the part before the S-BOX and the other one takes the responsibility for the calculation after the data passes through the S-BOX, but problems remain. Besides, it also causes multi-input problem and chaos inputs and enlarges the design space requirement. Also, through the analysis, the size of the S-box is the determinate to improve the encrypting performance. Applying the new static S-box designed by using the inversion and affine transformation in the AES encrypting system in short groups, the Anti-Square attacking ability performance of the system could be improved significantly. Also, the application of the new static S-box can increase the diffusivity of the system clearly. Give the condition of suited memory space and operating speed, changing the size of the S-box or the operating domain of the shift rows properly could reduce or eliminate the equilibrium while the Square attacking happening, and improve the Anti-Square attacking ability of the AES encrypting system in short groups and the security and the diffusivity of the AES algorithm.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure3: Example of S-box[12]

Phase 3: Segmentation and Validation Algorithm

The last but not the least one is to combining the AES algorithm with traditional data segmentation and validation algorithm that could validate the data size according to the input data size and increases the speed of the encryption and the decryption. Usually, if the test contains a plenty of data, the AES algorithm is used to encrypt and decrypt the test while if the test contains bigger data, the segmentation algorithm is applied prior to encrypt and decrypt the test. Therefore, the speed of the encryption and the decryption is fast and is close to the level-best AES algorithm speed. This mechanism has added the robustness and flexibility in the AES algorithm. Moreover, where the AES encryption carrying the encryption keys, the segmentation algorithm is used prior the encryption and after the AES decryption while transmission. For further improvement, the design may divide the nine rounds into three parts, which means every three rounds will be reputed as one block, and the three blocks will complete the whole nine rounds. This method is known as the pipeline that will increase the operating speed of the whole system. There will be no delay between any two blocks connected, and will save time for the data transmission.

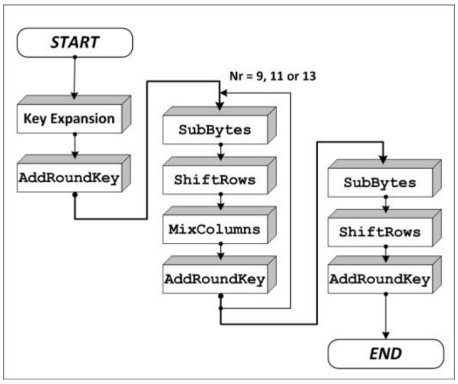

Figure4: The architecture of AES Algorithm [13]

Algorithm 1: The Rapid AES Algorithm

1. Input Data Matrix (d)
2. Data Matrix Validation → validate(d) → d_M

3. Data Matrix Segmentation → segment((d_M) → d_m^i
4. Input Security Key (S_k)
5. KeyExpansion(S_k)
6. InitialRound → AddRoundKey (S_k)
7. Rounds → For Loop
    a. *SubBytes(d_m^i)*
    b. *ShiftRows(d_m^i)*
    c. *MixColumns(d_m^i)*
    d. *AddRoundKey(d_m^i)*
8. Rounds → End For Loop
9. Final Round → MixColumns(*False*)
    a. *SubBytes(d_m^i)*
    b. *ShiftRows (d_m^i)*
    c. *AddRoundKey(d_m^i)*
10. Data Matrix Merger → merge(d_m^i) → d_M
11. Data Matrix Reverse validation → rvalidate(d_M) → d

With the help of above Scenarios ,we can encrypt and decrypt the images.In this implementation we take a set of 50 Different images(having different sizes).These Images are encryted with Existing AES and Rapid AES.and Calculate the Encryption and Decryption time ,Encryption and decryption speed . on the basis of these result we compare both the Existing and Rapid AES and draw the graph.

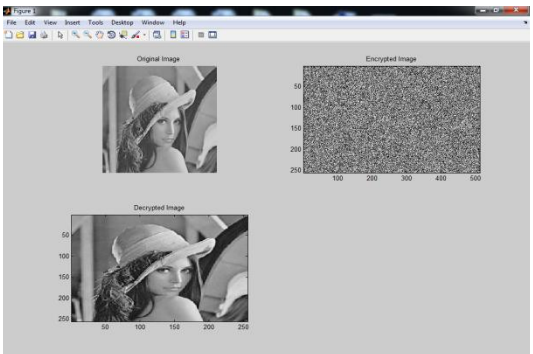

Figure5: Encryption and decryption of Image

EXPERIMENTAL RESULT

All measurements were taken on a single core of an Intel Core i3-2400 CPU at 3100 MHz, and averaged over 100000 repetitions. Our findings are summarized in Table 2 and Table 3. One can see that while the initialization overhead generally has a huge impact on the performance, this effect starts to fade out already at messages of around 256-1500 bytes. Due to the Rapid implementation of AES algorithm, it has performed way better than the existing AES encryption methods available. The Rapid algorithm achieves nearly optimal performance starting from 512 byte message length due to its ability of programming structure which enables it to fully utilize the improved multiple encryption patterns and validation for its initialization overhead. The Rapid AES algorithm has generally performs better than the existing when configured with block size of 128-bit and fixed s-box implementation. Also the validation method has been added to provide more flexibility and robustness to the proposed algorithm.

Index	File Size (Kb)	RAPID SCHEME		EXISTING SCHEME	
		Encryption Time (seconds)	Decryption Time (seconds)	Encryption Time (seconds)	Decryption Time (Seconds)
1	1183.711	2.565694	0.646096	11.60778	10.28958
2	811.6875	1.736718	0.625936	7.973028	7.549324
3	689.6484	1.454134	0.62148	7.106211	6.66421
4	585.7813	1.236788	0.626178	5.839042	5.249151
5	930.1094	1.958269	0.623342	9.336228	8.407667
6	1481.406	3.086175	0.622886	14.74685	13.03476
7	464.9531	0.988719	0.626737	4.742196	4.296639
8	1106.797	2.322694	0.62038	11.07452	10.25855
9	339.4688	0.724391	0.623164	3.458914	3.13864
10	793.4063	1.656799	0.626932	8.34238	7.231282
11	1347.328	2.822294	0.622072	13.83517	12.14879
12	988.3672	2.083992	0.624786	10.15554	8.96825
13	987.7813	2.076051	0.6259	10.25987	9.525794
14	837.7031	1.760459	0.624406	8.473982	7.558881
15	632.4609	1.333392	0.619688	6.39154	5.502444
16	1370.156	2.880993	0.623049	13.88179	12.39352
17	861.3281	1.818985	0.625489	8.544957	7.559276
18	1215.5	2.54242	0.623796	12.44276	10.8589
19	889.875	1.896367	0.62476	9.271283	8.295494
20	827.3828	1.740709	0.623639	8.228253	7.261873
21	1276.133	2.663857	0.622935	13.1867	11.80048
22	1403.5	2.94659	0.625573	14.31333	12.92009
23	1127.906	1.074278	0.622783	5.285584	5.416851
24	883.5469	1.860694	0.623106	10.16153	7.988927
25	1825.641	3.834191	0.625366	18.813	17.10785
26	1145.063	2.398937	0.623667	11.61872	10.07544
27	792.9297	1.675993	0.625483	8.936339	6.941237
28	797.0156	1.677833	0.626425	7.910779	7.030197
29	986	2.06346	0.625672	9.986034	9.198403
30	548.4375	1.157437	0.622987	5.257926	4.652692
31	535.6484	1.13789	0.6282	5.102753	4.525169
32	779.625	1.640933	0.622817	7.625147	6.769503
33	1275.984	2.69129	0.623454	12.33001	10.95515
34	969.7734	2.041494	0.623831	10.29159	8.720401

35	1332.203	2.814747	0.620471	13.78665	12.18908
36	511.875	1.082142	0.623492	5.107038	4.517749
37	986.1094	2.086753	0.629041	10.09526	8.823694
38	854.8125	1.79092	0.625686	8.553476	7.596591
39	886.8594	1.869049	0.631943	8.540838	7.583941
40	538.375	1.129189	0.626361	5.372788	4.764169
41	775.5313	1.620745	0.623829	7.637785	6.723918
42	632.0938	1.334828	0.623779	6.341472	5.629703
43	991.4063	2.078289	0.627298	9.987699	8.887296
44	800.0781	1.684204	0.623612	7.943403	7.030006
45	1177	2.452325	0.624992	11.96938	10.56235
46	815.5781	1.716208	0.623753	8.179212	7.258793
47	1342.523	2.808931	0.621929	13.41757	11.87779
48	1052.133	2.200702	0.622611	12.03852	9.23118
49	750.125	1.579631	0.625282	9.294883	6.710888
50	1271.406	2.65061	0.619143	12.49126	11.09007

Table2: The table displaying the results of Rapid AES implementation on dataset of 50 images (Encryption/Decryption Time)

The image dataset of 50 images of different sizes has been used for testing the performance of the Rapid algorithm. The rapid algorithm has been tested on all of the 50 images and the results have been represented in the table 2&3. The results have proved the effectiveness of the time proposed algorithm. The performance of the Rapid algorithm has been evaluated on the Index Core i3 CPU with 2GB RAM. The encrypted speed has been recorded between 450 and 500 Kbps. The average value recorded for the encryption module has been recorded at the 472 Kbps. The Decryption process has been recorded between 544 Kbps and 2858, whereas the average decryption speed has been recorded at 1474 Kbps speed. The encryption and decryption speeds have proved the effectiveness of the proposed algorithm on the image databases. The elapsed time for encryption process and decryption process have also been recorded. The average file of the image dataset, when converted to the double type has been recorded at 948 Kb.

\index	File Size (Kb)	RAPID Scheme		Existing Scheme	
		Encryption Speed (Kbps)	Decryption Speed (Kbps)	Encryption Speed (Kbps)	Decryption Speed (Kbps)
1	1183.711	461.361	1832.099	101.9757	115.0397
2	811.6875	467.3686	1296.759	101.8042	107.5179
3	689.6484	474.2674	1109.686	97.04869	103.4854
4	585.7813	473.631	935.4865	100.3215	111.5954
5	930.1094	474.965	1492.134	99.62368	110.6263
6	1481.406	480.0137	2378.294	100.4558	113.6504
7	464.9531	470.2579	741.8636	98.04595	108.2132
8	1106.797	476.5143	1784.063	99.94088	107.8902
9	339.4688	468.6262	544.7502	98.14317	108.1579
10	793.4063	478.8789	1265.539	95.1055	109.7186
11	1347.328	477.3875	2165.87	97.38426	110.9023
12	988.3672	474.2663	1581.929	97.32296	110.2074
13	987.7813	475.7982	1578.178	96.27624	103.6954
14	837.7031	475.8437	1341.6	98.8559	110.8237
15	632.4609	474.3247	1020.612	98.95282	114.9418
16	1370.156	475.5848	2199.115	98.70171	110.5543
17	861.3281	473.5213	1377.048	100.7996	113.9432
18	1215.5	478.0877	1948.553	97.68733	111.9358
19	889.875	469.2526	1424.348	95.98186	107.2721
20	827.3828	475.3136	1326.701	100.5539	113.9352
21	1276.133	479.0545	2048.581	96.77422	108.1424
22	1403.5	476.3133	2243.543	98.05544	108.6293
23	1127.906	476.5994	822.1156	96.86726	94.51986
24	883.5469	474.848	1417.972	86.95019	110.5964
25	1825.641	476.1475	2919.317	97.04143	106.7136
26	1145.063	477.3207	1836.017	98.55322	113.6489
27	792.9297	473.1103	1267.708	88.73093	114.2346
28	797.0156	475.0268	1272.323	100.7506	113.3703
29	986	477.8382	1575.905	98.7379	107.1925
30	548.4375	473.8379	880.335	104.3068	117.8753
31	535.6484	470.7383	852.6717	104.9724	118.3709
32	779.625	475.1108	1251.771	102.2439	115.1672
33	1275.984	474.1162	2046.638	103.4861	116.4735
34	969.7734	475.0313	1554.546	94.22968	111.2074
35	1332.203	473.2942	2147.084	96.62996	109.2948
36	511.875	473.0202	820.9814	100.2293	113.3031
37	986.1094	472.5567	1567.64	97.68048	111.757
38	854.8125	477.3036	1366.201	99.93744	112.5258
39	886.8594	474.4977	1403.384	103.8375	116.9391
40	538.375	476.7801	859.5288	100.204	113.005
41	775.5313	478.5029	1243.179	101.5388	115.3392
42	632.0938	473.5396	1013.33	99.67619	112.2784
43	991.4063	477.0301	1580.438	99.26273	111.5532
44	800.0781	475.0482	1282.974	100.7223	113.809
45	1177	479.9526	1883.223	98.33429	111.4335
46	815.5781	475.2209	1307.533	99.71354	112.3573
47	1342.523	477.9482	2158.645	100.0572	113.028
48	1052.133	478.0897	1689.871	87.39722	113.976

ISSN: 2278-0181
Vol. 3 Issue 10, October- 2014

49	750.125	474.8736	1199.659	80.703	111.7773
50	1271.406	479.6656	2053.493	101.7837	114.6437

Table3: The table displaying the results of Rapid AES implementation on dataset of 50 images (Encryption/Decryption Speed)

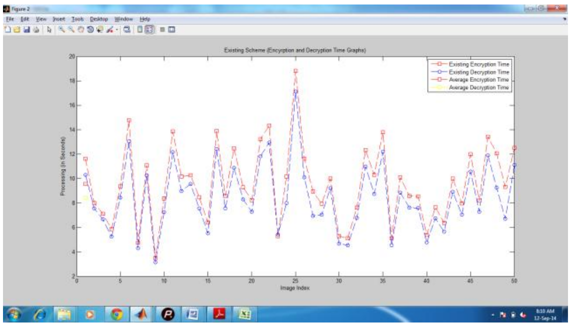

Figure6: The graphs of Encryption and Decryption time for existing system

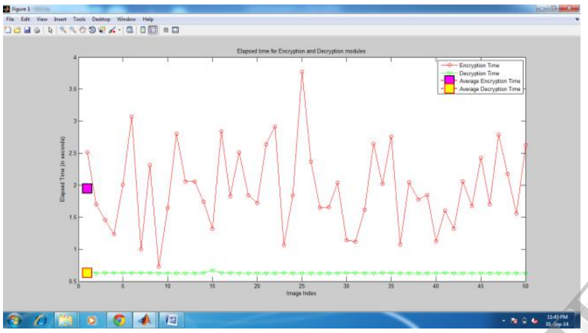

Figure7: The graphs of Encryption and Decryption time for Rapid system

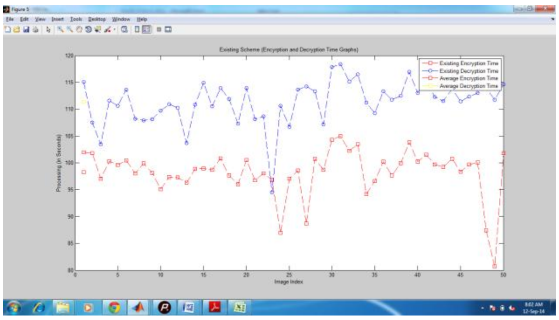

Figure8: The graphs of Encryption and Decryption processing speeds [also the average results] for existing model

The JPEG or JPG files saved on the disk are in the saved in the lossless compressed format, which is done to save the disk space on the user's gadget or PC. There are several variants of JPEG encryption are available now-a-days. The JPG or JPEG compression type uses discrete cosine transform or discrete wavelet transfer or their combination to

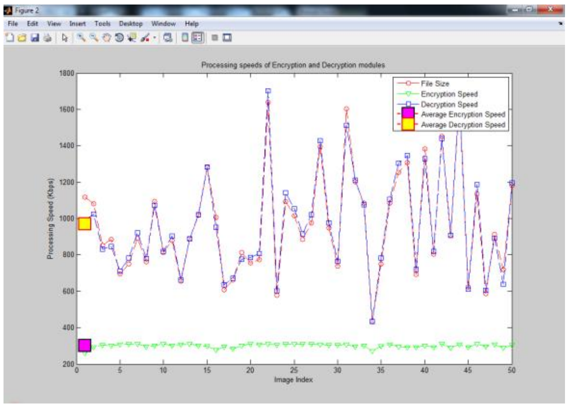

Figure9: The graphs of Encryption and Decryption processing speeds [also the average results] for Rapid model

Compress the data on the disk. When this image data is loaded into the memory, it is extracted to the actual size of the image data matrix. The average elapsed time of encryption module for all of the images in the image dataset has been recorded at 2 seconds. The average decryption time has been recorded at 0.64 seconds. These statistics have shown the effectiveness of the Rapid AES algorithm in the real-time picture. The sizes shown in the table 2 and 3 are the real sizes of the images stored on the disk. The performance of the proposed algorithm can have slight variations on each performance test because of the variation in the CPU usage and RAM usage on the PC due to operating system or other processes.

Average File Size	948 Kb
Average Encryption Time	2 seconds
Average Decryption Time	0.64 seconds
Average Encryption Speed	472 Kb/second
Average Decryption Speed	1474 Kb/second

Table4: displaying the mean of the results of Rapid AES implementation on dataset of 50 images

CONCLUSIONS

The Rapid AES has been deployed with static S-Box to minimize the effort to create S-Box on runtimes which consumes handful amount of time. The speed of the Rapid algorithm has been also improved by using various programming optimization methods. The last improvement has been made in the division of data into chunks according to algorithm block size. The data segmentation, validation and data aggregation algorithm has been designed in the way to perform faster than other existing options. The results have proved that Rapid AES has performed better than the existing AES on image data type. The Rapid and existing AES algorithms has been recorded for their encryption speeds, elapsed time for encryption, elapsed time for decryption, decryption speeds, etc. The proposed (Rapid) scheme has performed better on all of the fronts and has proved itself faster than the existing AES encryption algorithm.

FUTURE SCOPE

In the future, a survey on the Rapid AES scheme can be conducted to evaluate its performance on various types of data like video, audio, text, image, etc. Also the algorithm can be enhanced for the improvements in the speed, robustness or hardened security. Future researchers can take inspiration from the Rapid model to develop a new encryption paradigm.

REFERENCES

1. Gary C.Kessler, "An Overview of Cryptography: Cryptographic", *HLAN*, ver. 1, 1999-2014.
2. Navita Agarwal, Himanshu Sharma "An Efficient Pixel-shuffling Based Compression, Encryption and Steganography", *International Journal of Computer Science and Mobile Computing*, vol. 2 issue 5,pp. 376-385, May 2013.
3. Milind Mathur, Ayush Kesarwani, "Comparison between DES, 3DES, RC2, RC6, BLOWFISH and AES", in *National Conference on New Horizons in IT*, vol. 1, pp. 143-148, 2013.
4. Verma O.P., Agarwal R., Dafouti D., "Performance analysis of data encryption algorithms", in *International Conference on Electronics Computer Technology*, pp. 399-403, 2011.
5. Diaa Salama Abd Elminaam, Hatem Mohamed Abdual Kader, and Mohiy Mohamed Hadhoud, "Evaluating the Performance of Symmetric Encryption Algorithms", *International Journal of Network Security*, vol.10, no.3, pp.216–222, May 2010.
6. National Bureau of Standards, "Data Encryption Standard," FIPS Publication 46, 1977.
7. William Stallings "Network Security Essentials", Pearson Education, 2004
8. Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encryption Standard." March 2001.
9. Akanksha Mathur "A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms", *International Journal on Computer Science and Engineering*, vol. 4, no. 09, Sep 2012
10. Md. Nazrul Islam, Md. Monir Hossain Mia, Muhammad F. I. Chowdhury and M.A. Matin "Effect of Security Increment to Symmetric Data Encryption through AES Methodology" in *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 291-294, 2008
11. Majdi Al-qdah & Lin Yi Hui "Simple Encryption/Decryption Application"in *International Journal of Computer Science and Security*, vol 1, p.33,2007.
12. Gurjeevan Singh, Ashwani Kumar Singla, K.S. Sandha, "Throughput Analysis of Various Encryption Algorithms", *International Journal of Computer Science and Technology*, vol. 2, issue 3, September 2011.
13. Turki Al-Somani, Khalid Al-Zamil "Performance Evaluation of Three Encryption/Decryption Algorithms on the SunOS and Linux Operating Systems".