Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019)
IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4

Performance Evaluation of AES, ARC2, Blowfish, CAST and DES3 for Standalone Systems

Symmetric Keying Algorithms

Arpit Kubadia
Mukesh Patel School of Technology Management and Engineering, NMIMS
Mumbai, India
arpitkubadia23@gmail.com

Drishti Idnani
Mukesh Patel School of Technology Management and Engineering, NMIMS
Mumbai, India
drishti.idnani@gmail.com

Yash Jain
Mukesh Patel School of Technology Management and Engineering, NMIMS
Mumbai, India
yashchanchaljain@gmail.com

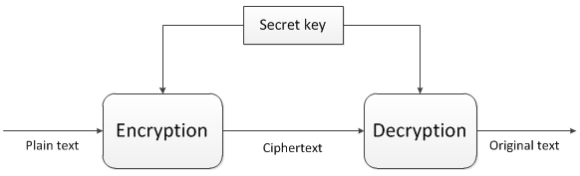
***Abstract*— The paper presents a comparative analysis of different symmetric keying cryptographic algorithms. The algorithms under study are AES, ARC2, Blowfish, CAST and DES3, and the performance metrics for the basis of the study are Encryption time, Decryption time, Throughput, Plain text size vs Cipher text size. The study is conducted for a standalone system, where the algorithms are executed on a quad core i5 processor running Windows 10 and, Python is used as the programming language.**

Keywords-Encryption, AES, ARC2, Blowfish, CAST, DES3, symmetric encryption, Python, Standalone-system

I. INTRODUCTION

The first cryptanalytic documentation dates back to AD 800 where Al-Kindi from Arabia tried to break monoalphabetic substitution ciphers. The second milestone in the field of cryptography was the use of German Enigma coding during the World War 2 that gave the Nazi Germany force a significant advantage over the others, though later the cipher was broken. The importance of data protection has increased exponentially since then supported majorly by digitalization. The modern algorithms use key/keys(public or private) to encrypt and decrypt data for secure transmission. The number of keys and the crux of encryption depends on whether it is a symmetric keying or a non-symmetric keying algorithm. Since so many algorithms have been developed right from DES(Data Encryption Standards) and AES(Advanced Encryption Standards) for initial use over the internet, to the most recent Blowfish and 3DES, we need to perform an analysis study to find the best suited algorithm based on different performance parameters.
Symmetric key works with a single sender and receiver key whereas Asymmetric has a public key for encryption and a private key for decryption, which is unique for a user.

The parameters that affect the efficiency of cryptographic algorithms are as follows:



1. **Encryption Time:** The time taken by the algorithm to convert the plain text to cipher text.
2. **Decryption Time**: Time required to extract the original text from the cipher text.
3. **Throughput:** Number of kilo-bytes(KB) of code processed per second (KB/sec).
4. **Plain text size vs Cipher text size:** The change in text size due to padding for encryption or logical operations.

II. LITERARY SURVEY

The performance of any algorithm depends upon the configuration of the system on which it is executed and also on the availability of the Processor memory(i.e. how many other processes are occupying the CPU).
Therefore we have used the following configuration to conduct the performance study:

1. 8 GB RAM
2. Intel core i5 processor
3. Quad core system clocking at 1.8 GHz(min) to 3.39 GHz (max)
4. 64 bit Windows 10 operating system.

All other processes were forcefully shut in order to provide the entire CPU power to the algorithm.
The average CPU consumption during execution was found to be 3.62%.

A review of various papers on cryptography showed that symmetric encryption is nearly 1000 folds faster than Asymmetric encryption and requires a comparatively smaller key size.[1] Hence it is practical to analyse symmetric key algorithms their performance metrics.

The following algorithms were found to be comparative and most used based on different paper reviews, hence the considerations:

Sr. No	*Algorithm*	*Type*	*Block size*
1.	AES	Block	128 bits
2.	ARC2	Block	64 bit
3.	Blowfish	Block	64 bit
4.	CAST	Block	128 bit
5.	DES3	Block	64 bit

Fig. 1. Symmetric keying algorithms

III. OBJECTIVE

The objectives of the paper is to conclude a best suited symmetric cryptography algorithm based on standard evaluation metrics for a standalone system.

IV. SIMULATION PROCEDURE

Our goal is to measure the Encryption time, Decryption time and Throughput of the algorithms to indicate its speed, along with the size of cipher text vs the size of plain text to give the algorithm efficiency. The algorithms are implemented on python due to availability of different libraries required to run the tests.

Time calculation was done using timestamping before and after encryption and decryption; the difference of which gave the total time respectively. The cipher text size was extracted from the output data by string manipulation.

V. DETAILED DESCRIPTION OF ENCRYPTION ALGORITHMS

1. AES

Advanced Encryption Standards was recommended by NIST which was brought in use in place of DES in the year 2001. AES is a symmetric block cipher that supports various combination of data(128 bit) and key size ranging from 128 to 256.

The number of rounds required for its completion is as follows:[2]

Key Size (bits)	Rounds
128	10
192	12
256	14

Fig. 2. Rounds required

During each round the data goes through the following changes:

- Sub-bytes
- Shift-rows
- Mix-columns
- Add round key

Message Size(MB)	Encryption time (sec)	Decryption time(sec)	Cipher text size (MB)	Throughput (KB/sec)
1	0.033806	0.050442	1.564571	12154.5912
2	0.087838	0.089903	3.129139	11522.3837
4	0.165196	0.160118	6.258274	12590.9121
8	0.316141	0.318748	12.516544	12903.0429
16	0.647923	0.649272	25.033084	12630.3293
32	1.289156	1.293616	50.066165	12687.1438
64	2.659563	2.59036	100.132328	12483.2307
128	5.182357	5.174406	200.264652	12655.6917

Fig. 3. AES Performance for different message sizes

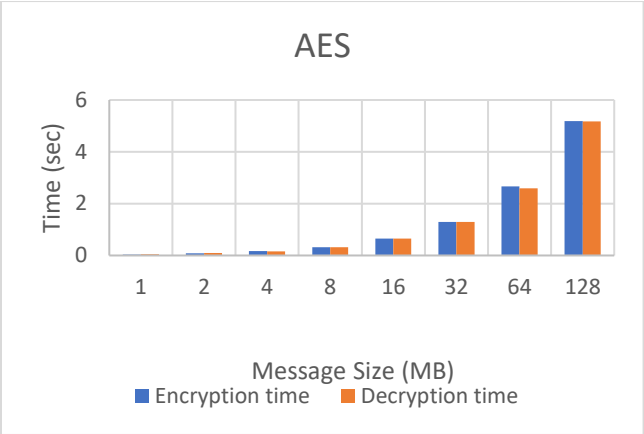


Fig. 4. Comparison of Encryption time and Decryption time

2. ARC2

Also known as RC2, it was an algorithm designed by Ron Rivest Security during 1987-1988.The key length varies from one byte to 128 bytes and the block size is of 64 bits. Each data block has 4 words r0,r1,r2,r3 which are modified and returned in the same array.[i]

The key is first expanded up to a byte limit given by

$$Key_{byte\text{-}limit} = (Key_{bit\text{-}limit} + 7) / 8 \quad (1)$$

Then a bit mask is implemented for assurance

$$Key_{mask} = 255 \bmod 2\text{^}(8 + Key_{bit\text{-}limit} - 8 \cdot Key_{byte\text{-}limit}) \quad (2)$$

Message Size(MB)	Encryption time (sec)	Decryption time(sec)	Cipher text size (MB)	Throughput (KB/sec)
1	0.301048	0.316168	1.564571	1659.06263
2	0.613953	0613953	3.129139	1672.48115
4	1.219824	1.217958	6.258274	168021586
8	2.495453	2.458589	12.516544	1653.59922
16	5.053597	5.110347	25.033084	1611.97267
32	9.797324	9.810896	50.066165	1671.13588
64	19.852633	19.577693	100.132328	1662.07096
128	39.329631	39.182846	200.264652	1669.44166

Fig. 5. ARC2 Performance for different message sizes

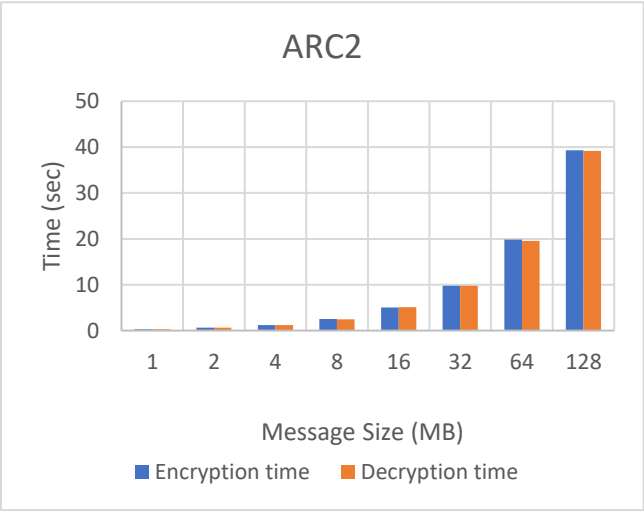


Fig. 6. Comparison of Encryption time and Decryption time

3. Blowfish

Designed by **Bruce Schneier** in 1993, it was a replacement that was faster and more reliable than the existing DES or others. It has a 64 bit block cipher and works with variable key length ranging from 32 bits to 448 bits.

Each *encryption* is a 16 round Feistel cipher and uses a key-dependant S-boxes.
Every round has 4 actions associated with it:
1. XOR left half bits with kth P-array entry
2. Input the XOR'ed data to blowfish function
3. XOR the output with the right half bits
4. Swap left and right part

Decryption is the same except that the P1 to P18 are used in reverse order.Abbreviations and Acronyms

Message Size(MB)	Encryption time (sec)	Decryption time(sec)	Cipher text size (MB)	Throughput (KB/sec)
1	0.032238	0.016798	1.564582	20882.6168
2	0.033161	0.034593	3.129150	30226.9976
4	0.082184	0.073582	6.258285	26295.8540
8	0.137474	0.137718	12.516555	29768.3072
16	0.279742	0.262489	25.033096	30215.9043
32	0.580346	0.560974	50.066177	28710.6157
64	1.160837	1.080431	100.132339	29240.5906
128	2.502643	2.186614	200.264663	27951.5496

Fig. 7. Blowfish Performance for different message sizes

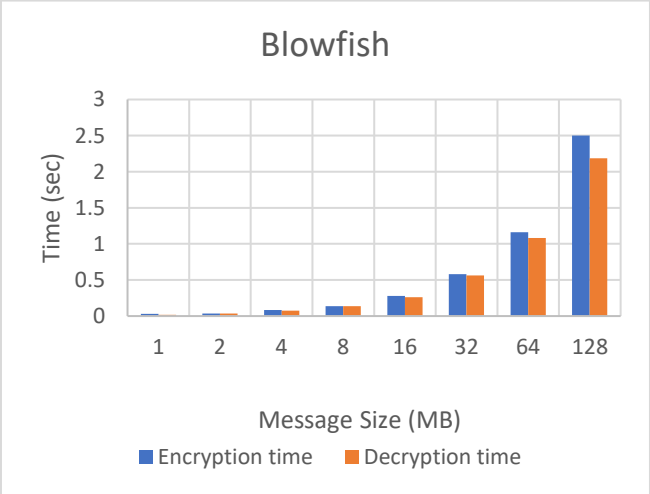


Fig. 8. Comparison of Encryption time and Decryption time

4. CAST

A symmetric key algorithm developed by Carlisle Adams and Stafford Tavares in 1996. It is under use by the Government of Canada Communications Security Establishment. The block size is 64 bit whereas the key size ranges from 40 to 128 bits. It makes use of the Feistel Network Structure and runs for 12 or 16 rounds depending on the application. 16 rounds are used when key size is greater than 80 bits.

Message Size(MB)	Encryption time (sec)	Decryption time(sec)	Cipher text size (MB)	Throughput (KB/sec)
1	0.144317	0.171853	1.564571	3238.76395
2	0.298107	0.29833	3.129138	3433.72393
4	0.56787	0.57122	6.258274	3595.85283
8	1.151932	1.139442	12.5165443	3575.14748
16	2.303868	2.28345	25.033084	3571.58583
32	4.566849	4.568915	50.066165	3586.78267
64	9.126131	9.135119	100.132328	3588.80142
128	18.330283	18.330419	200.264652	3575.27250

Fig. 9. CAST Performance for different message size

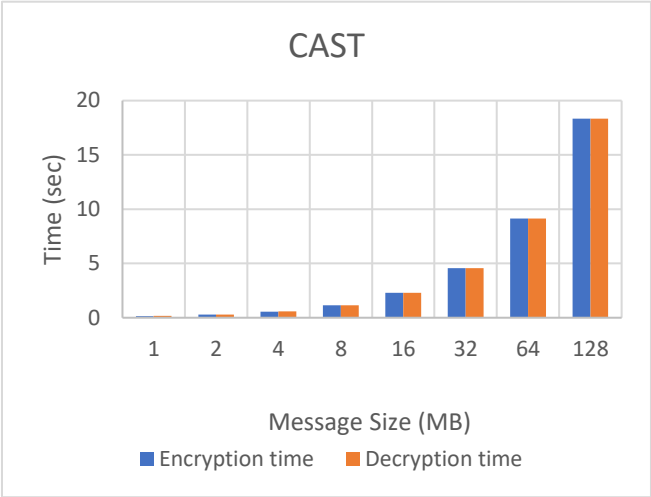


Fig. 10. Comparison of Encryption time and Decryption time

5. DES3

A symmetric block algorithm that applies the DES(Data Encryption Standards) technique three times to each data block. The block size is of 64 bit and the key sizes are 168 bit with keying option 1, 112bit with keying option 2 and 56bit with keying option 3.It runs for 48 DES equivalent rounds to complete the encoding or decoding.

Message Size(MB)	Encryption time (sec)	Decryption time(sec)	Cipher text size (MB)	Throughput (KB/sec)
1	0.08358	0.068857	1.564571	6717.52920
2	0.115017	0.152845	3.129138	7645.72802
4	0.267549	0.247588	6.258274	7951.28286
8	0.510196	0.51083	12.5165443	8023.30205
16	1.016141	0.998178	25.033084	8133.7663
32	1.976154	1.991167	50.066165	8259.47787
64	3.981097	3.999742	100.132328	8211.66797
128	7.981955	7.983601	200.264652	8209.67337

Fig. 11. CAST Performance for different message sizes

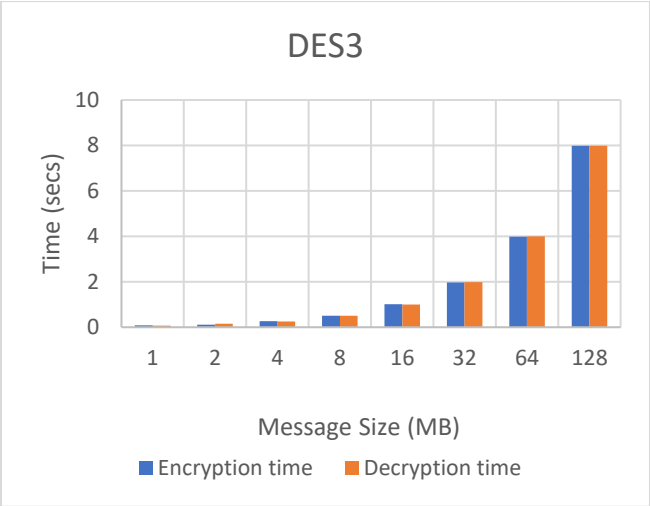


Fig. 12. Comparison of Encryption time and Decryption time

VI. RESULT AND DISCUSSION

The results are based on the comparative study of three different metrics, of all the 5 algorithms under the stated conditions viz.:

· The encryption time of the algorithms
· The decryption time of the algorithms
· The throughput of the algorithms
· Cipher to Text Ratio

1. Encryption time

Message Size	AES	ARC2	Blowfish	CAST	DES3
1	0.033806	0.301048	0.032238	0.144317	0.08358
2	0.087838	0.613953	0.033161	0.298107	0.115017
4	0.165196	1.219824	0.082184	0.56787	0.267549
8	0.316141	2.495453	0.137474	1.151932	0.510196
16	0.647923	5.053597	0.279742	2.303868	1.016141
32	1.289156	9.797324	0.580346	4.566849	1.976154
64	2.659563	19.852633	1.160837	9.126131	3.981097
128	5.182357	39.329631	2.502643	18.330283	7.981955

Fig. 13. Encryption time with varying message size

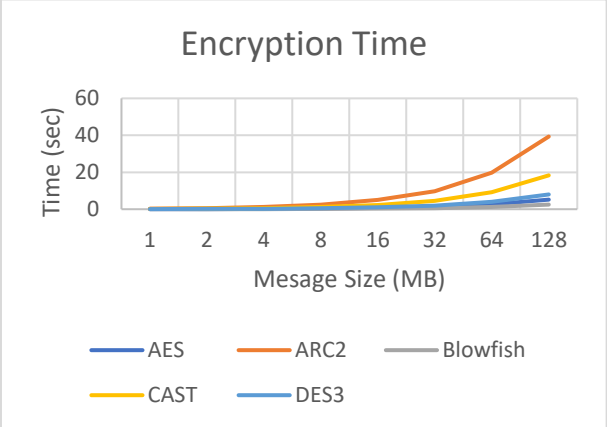


Fig. 14. Graph for encryption time with varying message size

From the plot it can be concluded that:

1. There is no significant variation in the encryption time when the text size ranges from 1MB to 8MB.
2. Beyond 8MB, ARC2 shows an exponential growth in encryption time complexity which is not desirable.
3. CAST shows slight exponential characteristics, yet it performs better than ARC2.
4. The best performance is shown by Blowfish which gives lowest variation in encryption time.
5. For 128 MB Blowfish is 1571.524% faster than ARC2, which is the worst performing algorithm, and 207.076% faster than AES, which is the best performing algorithm in the given set, excluding Blowfish.

2. Decryption time

Message Size	AES	ARC2	Blowfish	CAST	DES3
1	0.050442	0.316168	0.016798	0.171853	0.068857
2	0.089903	0.610575	0.034593	0.29833	0.152845
4	0.160118	1.217958	0.073582	0.57122	0.247588
8	0.318748	2.458589	0.137718	1.139442	0.51083
16	0.649272	5.110347	0.262489	2.28345	0.998178
32	1.293616	9.810896	0.560974	4.568915	1.991167
64	2.59036	19.577693	1.080431	9.135119	3.999742
128	5.174406	39.182846	2.186614	18.330419	7.983601

Fig. 15. Decryption time with varying message size

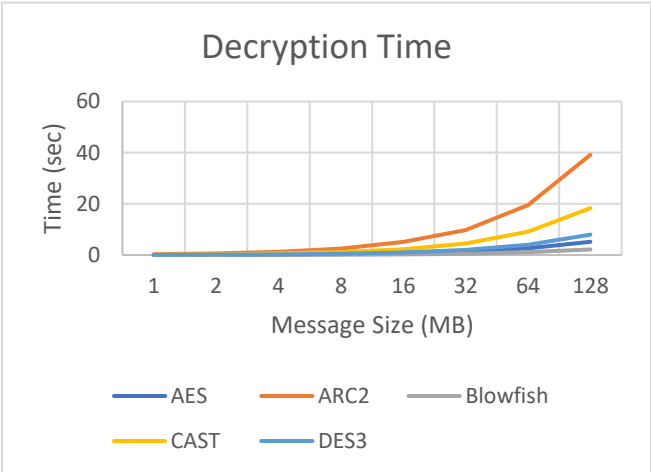


Fig. 16. Graph for decryption time with varying message size

The following conclusions can be drawn from the plot:

1. There is no significant variation in the decryption time when the text size ranges from 1MB to 8MB.

2. Beyond 8MB, ARC2 shows an exponential growth in encryption time complexity which is not desirable.
3. CAST shows slight exponential characteristics, yet it performs better than ARC2.
4. The best performance is shown by Blowfish which gives lowest variation in encryption time.
5. For 128 MB Blowfish is 1791.993% faster than ARC2, which is the worst performing algorithm, and 236.640% faster than AES, which is the best performing algorithm in the given set, excluding Blowfish.

The reason why Decryption time gives characteristics similar to that of encryption is that the algorithms follow symmetric keying which has the same algorithm for decryption, as encryption, in reverse order. The variation is due to changing of internal processing conditions if any.

3. Throughput

Msg Size	AES	ARC2	Blowfish	CAST	DES3
1	12154.591	1659.062	20882.616	3238.763	6717.529
2	11522.383	1672.481	30226.997	3433.723	7645.728
4	12590.912	1680.215	26295.854	3595.853	7951.282
8	12903.042	1653.599	29768.307	3575.147	8023.302
16	12630.329	1611.972	30215.904	3571.585	8133.766
32	12687.143	1671.135	28710.615	3586.782	8259.477
64	12483.230	1662.070	29240.590	3588.801	8211.667
128	12655.691	1669.441	27951.549	3575.272	8209.673

Fig. 17. Throughput with varying message size

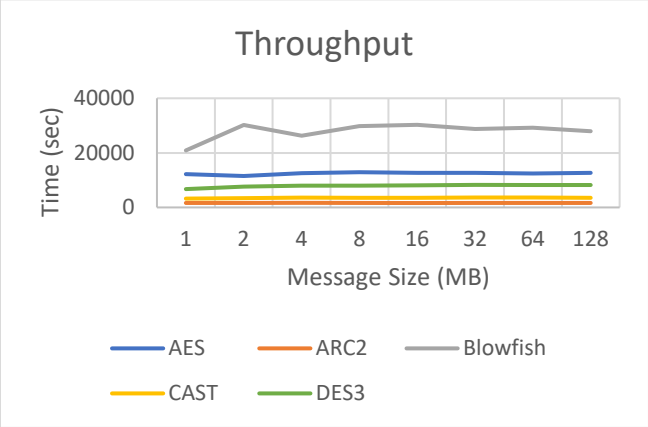


Fig. 18. Graph for throughput with varying message size

The plot displays the following characteristics:

1. All the algorithms except Blowfish deliver low but almost constant throughput.
2. The high throughput of the Blowfish is theoretically appreciable, but it has a lot of variation which is not desirable.
3. All factors considered, AES gives a high and constant throughput, but in applications where variation in time is not an affecting factor Blowfish provides the most efficient performance.

4. Cipher to Text Ratio

Message Size	AES	ARC2	Blowfish	CAST	DES3
1	1.564571381	1.564571381	1.564582825	1.564571381	1.564571381
2	1.564569473	1.564569473	1.564575195	1.564569473	1.564569473
4	1.56456852	1.56456852	1.564571381	1.56456852	1.56456852
8	1.564568043	1.564568043	1.564569473	1.564568043	1.564568043
16	1.564567804	1.564567804	1.56456852	1.564567804	1.564567804
32	1.564567685	1.564567685	1.564568043	1.564567685	1.564567685
64	1.564567626	1.564567626	1.564567804	1.564567626	1.564567626
128	1.564567596	1.564567596	1.564567685	1.564567596	1.564567596

Fig. 18. Cipher to text ratio with varying message size

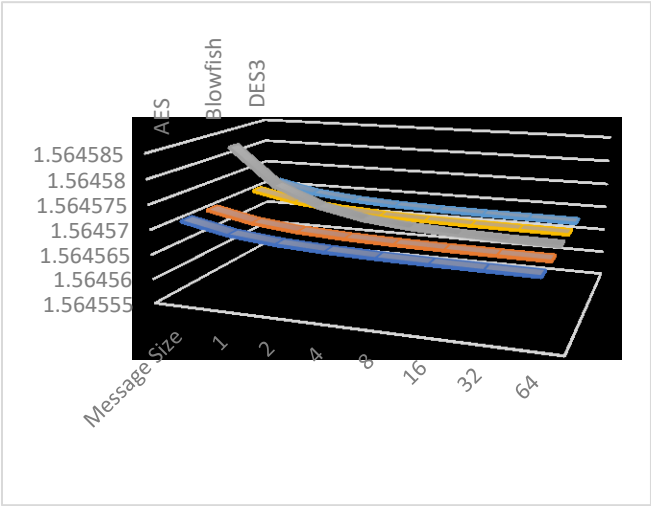


Fig. 19. Graph for cipher to text ratio with varying message size

The increase in size of the cipher text as compared to plain text is due to the padding generated by the algorithms for encryption.

The notable features from the cipher text to plain text ratio plot are:

1. The ratio decreases as the size of plain text decreases for all the algorithms under study.
2. The ratio for Blowfish is undesirably high when the plain text size is less than 8MB.
3. Above a threshold value of 16MB all the algorithms give same performance in relation to cipher text vs plain text size.

VII. CONCLUSION

The following conclusions are drawn from the study:

1. Compression of plaintext is needed before it is fed to any of the symmetric key algorithm to increase the memory efficiency.
2. Blowfish is the fastest algorithm but produces undesirable variation in throughput for lower plaintext size.
3. AES provides an overall consistent performance, marginally better than the others except Blowfish.
4. AES has a larger block size of 128 bit as compared to Blowfish's 64 bit, making it more secure against birthday attacks.

VIII. FUTURE SCOPE

The work produced herein has immense scope in the future. The primary aim of encryption is to maintain data integrity and confidentiality. Another major factor that measures the efficiency of cryptic algorithms is the avalanche effect which is a desired characteristic. If a cipher does not exhibit the avalanche effect it becomes easier for a cryptanalyst to break the cipher.

Hence the proposed future work shall include:

·Testing of algorithms for integrity against brute force attacks.

·Comparing the avalanche effect for algorithms.

·Extend the study to cloud and distributed systems.

IX. REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.