Select Symmetric Cryptographic Algorithms' Evaluation for Speed and Throughput

S. Deepak Raj[1] Pruthvi kumar K .R[2]
[1]Asst. Prof , Dept Of ISE, Sai Vidya Institute of Technology, Bangalore, India
[2]Asst. Prof , Dept Of CSE, Sai Vidya Institute of Technology, Bangalore, India

Deepak M[3] Harisha D S[4]
[3]Asst. Prof , Dept Of CSE, Sai Vidya Institute of Technology, Bangalore, India
[4]Asst. Prof , Dept Of ISE, Sai Vidya Institute of Technology, Bangalore, India

Abstract--Secure communication requires authentication, privacy, integrity and non-repudiation to be meaningful and successful. These requirements have always been there for quality communication. They have gained more importance than ever before, due to technological advancements in digital communication, in Internet and in network related communication applications. As the number of channels, intermediate entities and layers between end nodes increase, the need for secured communication also increases. A widely used technique to protect communication is encryption. Such techniques are implemented in algorithms called encryption algorithms. These algorithms that are used to encrypt communication data are identified and classified based on two parameters, their innate ability to encrypt and hence secure data against attacks and the speed with which they accomplish the task of encryption. Every algorithm on running makes use of system resources like memory, CPU time and Input Output devices. A study of such resource usage gives us insight into algorithm's relative efficiency. This paper compares six such popular algorithms, AES, DES, 3DES, RC2, RC6 and BLOWFISH considering time consumption or speed and throughput for encryption of varying input sizes.

Keywords--*Cryptography, Cryptographic algorithms, CPU Time, Throughput*

1. INTRODUCTION

Communication over internet requires us to communicate in such a way that none other than the sender and the receiver gets to read the contents of communication. Such secrecy of exchange can be achieved by a method called encryption. It is the process of converting a plain text in to an encrypted form called the cipher text. A host of encryption schemes or algorithms are presently available. They make use of a key to encrypt data. Key used is generated by the algorithm based on the extant of security required and on the input size. Once encrypted at the sender's end of the channel, only a user who has the corresponding key to decrypt the message can decipher the encrypted data [1]. There are two main variations in use of keys. If a single key is used for encryption as well as decryption, such a scheme is called symmetric key cryptography. If separate keys are used, the scheme is then called public key cryptography. Encryption is a means of conversing in such a manner that though eavesdropping occurs, the eavesdropper does not comprehend the contents of the message being communicated [2]. Public key encryption is mathematically intensive and found to be more secure [3]. The drawback of public key cryptographic algorithms when compared to symmetric methods is they tend to be thousand times slower; as they require more computational processing time [3]. Behrouz A. Forouzan et.al have in [17] made a prediction that the two broad classes on encipherment, symmetric as well as asymmetric will continue to co-exist and both methods will get rich with newer algorithms for considerable amount of time. The authors express their understanding further and explain the special feature of this dualistic approach to encryption. By the very nature of symmetry of encryption in the former and the lack of it in the later, they are logically inverses of each other. Approach to implementation is opposing in steps of both encryption and decryption. Due to these reasons, disadvantages of symmetric approach are overcome by using asymmetric approach and flaws of asymmetric methods are tackled using symmetric algorithms. There is also a trend to combine both the methods to find a synthetic mode of encryption. Conceptually, the vital difference among the two lies in how they create and share a piece of secret information. In symmetric methods, the secret information that is generated is shared among the two parties involved in communication. On the other hand, the generated secret information is maintained as a personal secret and not even shared among the parties involved in communication. Following this scheme requires generation of n (n-1)/2 number of shared secret pieces of information, also called keys if n people are communicating in a communication process. All the chosen algorithms are coded in C language and test executions are made in this work.

1.1AES Algorithm

AES was developed by two Belgian cryptographers Joan Daemen and Vincent Rijmen in 2001. It is now a specification of electronic data established by the United States National Institute of standards and Technologies (NIST). The algorithm is based on Rijndael cipher. Rijndael is a substitution linear transformation cipher. It does not require a Feistel network. Three discreet invertible uniform transformations is the idea implemented. The

transformations are linear mix, transform, non-linear transform and key addition transform. To enhance security, a key addition layer is performed at the very beginning. A state is conceptually formed when the algorithm begins execution but ends just before the end of encipherment. Such a state can be visualised as an array with 4 rows and column number being the block length divided by the bit length. AES is a block cipher private key algorithm that makes use of 128, 192 or 256 bits keys though the default size is 256. This scheme encrypts data blocks of 128 bits in 10, 12 and 14 rounds depending on the chosen key size. Striking features of this algorithm are its speed, flexibility and the ease with which it can be implemented on various platforms. An added advantage is its suitability to relatively small devices as shown by K. Nayak et.al in [5]. AES has also been tested extensively and found to perform under many applications of security in [6] by W. Stallings and by Daemen. J in [7]. Brute force attack is the only known attack against AES.

1.2 DES Algorithm

Data Encryption Standard was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). Its origin can be traced to IBM in the year 1977. It was later adopted by the United State Department of Defence. Its specification can be found in ANSI X3.92 and X3.106 standard in Federal FIPS 46 and 81 standards. Results from web source [18] show that over seventy two quadrillion keys can be generated in DES, making it a fairly popular algorithm. The algorithm contains 16 rounds with series of substitution and permutation. It is based on Lucifer algorithm proposed by IBM. The method makes use of 64 bit key size and 64 bit block size. Many attacks have been recorded and the algorithm's many weakness have shown it to be quite an insecure cipher by W. Stallings in [6], and by Coppersmith. D in [8].

1.3 *3DES Algorithm*

3DES is the popular acronym for triple DES. It is an extended version of Data Encryption Standard, which involves application of DES three times over every block of input data. It is a 64 bit block size with 192 bit key size algorithm. Study results from the web source [4] shows that 3DES in fact consumes almost thrice the time utilised by DES on the same input data.

1.4 *RC2*

RC2 makes use of variable lengths ranging from 8 bits to 128 bits. The block size is 64 bits. It was designed by Ron Rivest in 1987. RC in the name stands for "Ron's Code" or "Rivest's Cipher". Its eighteen rounds are arranged in a source heavy Feistel network with fifteen rounds of one type punctuated with two rounds of another type. It is shown in [12] by J. Kelsey et.al that RC2 is vulnerable to a related key attack using 2^{34} chosen plain texts.

1.5 RC6

RC6 was designed by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin. It is a block cipher that was derived from RC5 to be compliant to Advanced Encryption Standard (AES). It supports varied key sizes of 128 bits, 192 bits or 256 bits. N. El-Fishawy, in [14] has listed convincible reasons to consider RC6 as AES itself. RC6 does use an extra multiplication operation which is not present in RC5 in order to make the rotation dependent on every bit in a word, and not just the least significant few bits.

1.5 *Blow Fish*

The original Blowfish paper was presented at the First, Fast Software Encryption workshop in Cambridge, United Kingdom. Blow Fish was designed by Bruce Schneier in 1993. It is keyed symmetric block cipher. It was built as a general purpose algorithm to replace DES. It has a variable key length. Its block size is 64 bits. Aamer Nadeem et.al in [15] and Bruce Schneier et.al in [16] have shown that though Blow Fish suffers from weak key attacks, no attack has been successful against it. It has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm. Blowfish is unpatented and license-free, and is available free for all uses. Many cryptographers have examined Blowfish, although there are few published results as shown by the web source [19].

Figure.1 below shows a schematic representation of encryption algorithm classification. The schematic is only a graphical representation of classification. Many more algorithms not shown here do exist under the subclasses.

As shown by Diaasalama et.al in [13], encryption algorithms make use of considerable quantities of system resources namely, CPU time, memory and battery power. This work chooses throughput and time efficiency for evaluation. Shifting, floating point operations, logical operations at bit level as well of extensive XOR are characteristic requirements of both encryption as well as decryption. Every operation or set of operations that forms the heart of an encryption/decryption scheme is required to be applied repeatedly to every constituent block of the input block. Greater the fragmentation of input into blocks, greater is the load on the system resources in terms of operation overload. Considering this fact, inferences on relative advantages or disadvantages of using algorithms directly depends upon the amount of system resource and CPU time utilised.

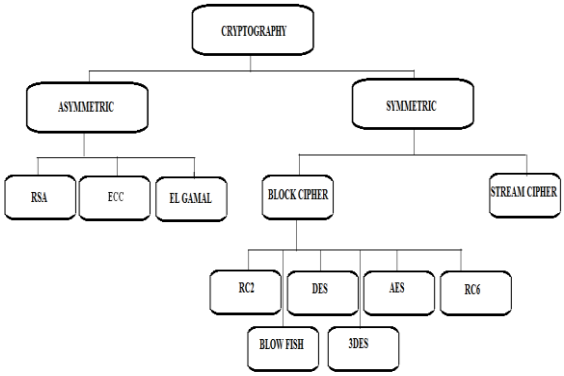

Figure.1 Encryption algorithms classified

Section 2 of this paper deals with related work. It makes a note of works that have been studied under literature survey and explains the reason behind choosing our experimental parameters. Section 3 explains the encryption model designed for evaluation. Section 4 presents the results in a consolidated, graphical format and explains the analysis. Finally the section 5 gives the conclusions followed by references.

2. RELATED WORK

S. Hirani has explained in [9] that AES is both fast and efficient compared to other algorithms. Since his work is limited to wireless devices, treatment of encryption is stand-alone but in limited perspective of their implementation in a single class of devices. Study in [10] by A. Nadeem has made an analysis of popular algorithms DES, 3DES, AES and Blowfish. This study is a performance analysis of algorithms in terms of input and nature of inputs. Input files of varying contents and sizes have been used as inputs. P. Ruangachari et.al in[11] describe energy consumption of different symmetric key encryptions for hand held devices. Again the drawback is a self imposed limitation of algorithm performance to wireless LAN's only. Considering suggestions for future work and recognizing areas which have not been dealt with in previously carried studies, this work chooses AES, DES, 3DES, RC2, RC6 and BLOWFISH for evaluation. On analysing available literature, purpose of this work is decided to make the study of algorithms as generic as possible to ensure that results from experiments will objectively compare algorithms.

3. ENCRYPTION MODEL

The system configuration used for the evaluation is Intel® Core™ Duo(2), 781MHz and 2.8 GHz CPU 3.45 GB RAM. Seven fixed size inputs are used to run tests on AES, DES, 3DES, RC2, RC6 and Blow Fish. Execution result valuations are graphed for time consumed for encryption and average throughput. Time is noted in milliseconds and memory in KB. Time consumed is the time taken to produce the encrypted output from the instant a call is made to the encryption function. Using this observation the derived parameter throughput can be easily computed for comparisons. An average value of input size is computed as average work load = $(\sum_{i=a}^{g} I)/n$ where, I is the input size of the file in KB, whose number n ranges over a to g held in the variable i. The average work load is then divided by the time consumed by each individual algorithm over the entire test execution. Observed results are graphed to compare relative performance as well as growth of performance curve. Computed results are graphed as a single relative growth curve to aid in direct comparison of performance in terms of throughput.

4. RESULTS

Experimental results have been presented here in the form of graphs. Input A corresponds to data size of 50 KB, Input B to 60 KB, Input C to 100 KB, Input D to 210 KB, Input E to 310 KB, Input F to 700 KB and Input G to 900 KB. For each of the seven input sizes, all the six algorithms AES, DES, 3DES, RC2, RC6 and Blow Fish are executed and readings of running times are measured. Y-axis in the figure.1 marked with time consumption values in milliseconds and with throughput in the second. Figure.1 charts performance of algorithms in terms of time consumed for input A, B, C, D, E, F and G. RC2 is observed to have utilized maximum time. Blow Fish is observed to have utilized minimum time and AES, DES, 3DES and RC6 is observed to have utilized intermediate amount of time. This trend deviates only in case of input E over a minimum value of decimal or fractional part. The graph in Figure.2, charts the algorithms' behaviour in terms of average computed throughput. Throughput of Blow Fish is found to be maximum, that of AES is minimum. DES and RC6 are of comparable throughputs. Throughput of RC2 is less than 3DES. Speed of RC6 is faster than the highest average speed algorithm Blow Fish at input A and input D. At all other inputs, this behaviour deviates. At input E, both DES and 3DES perform better than Blow Fish in terms of speed, again this behaviour is localised to the input in consideration, the behaviour deviates sharply beyond input E. In case of the maximum time consumption and hence minimum speed however, RC2 is a tough contender with no other algorithm superseding its records for all inputs from A to G. For inputs between C and D, a peculiar behaviour of algorithms, AES, DES, 3DES, RC6 and Blow Fish is observed. Plainly, all chosen algorithms excluding, RC2 are observed in this region of input to perform very close to each other in terms of speed. The curves of graphs can be seen to almost overlap and form a dense band of five lines. Another striking observation of all algorithms in terms of variance of speed can be observed in the relative change in slopes of the graphs. Slopes of graphs appear very close in magnitude between the inputs A and B, between B and C and again between E and F. But the most similar variation of slope is at the last range of inputs that is between F and G. This behaviour varies only in the intermediate range of graphs between C and D and D and E respectively. 3DES has the most stable growth of graph in terms of slope.

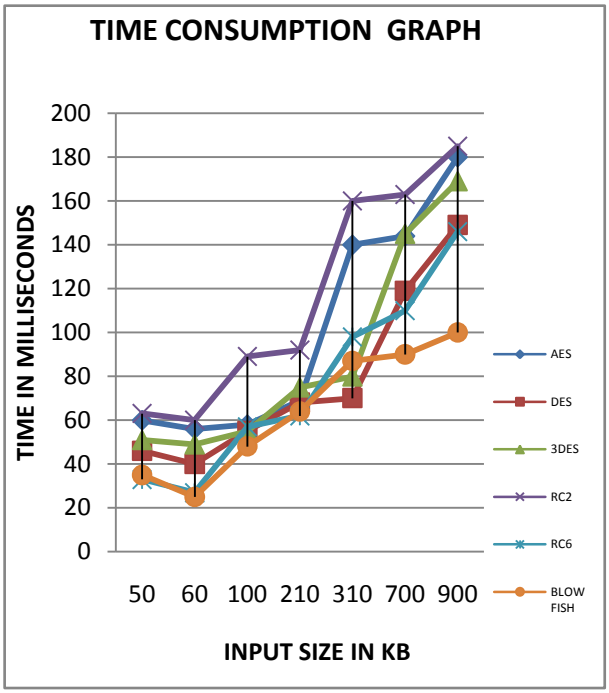

Figure.1 Results charted for time consumption

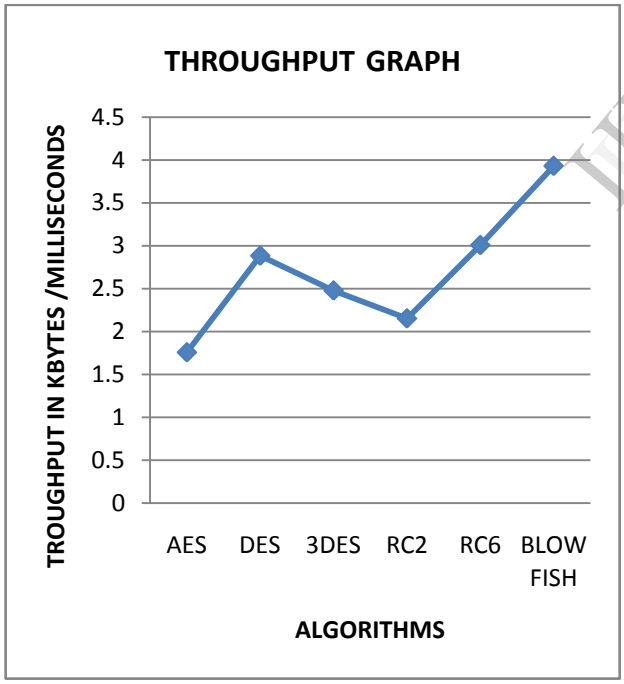

Fig.2 Results charted for relative throughput

5. CONCLUSION

Efficiency of encryption algorithms can be computed and two or more algorithms can be compared using their speed, resource utilization and robustness. Here AES, DES, 3DES, RC2, RC6 and Blow Fish have been considered for analysis and they are compared in terms of computation time and average throughput. RC2 is found to consume maximum time compared to AES, DES, 3DES, RC6 and Blow Fish. Blow fish is found to be better than all others in terms of time consumed. In terms of throughput, AES is minimum and Blow Fish is maximum. It can be clearly observed from the graph in Figure.1, and an inference can be drawn that time consumption gradually increases as the input size increases with the behaviour of DES being erratic. We cannot find an implicative reason for the phenomenon since the nature of inputs and actual contents of inputs have not been considered in designing the work load. Algorithms chosen in this work treat inputs as blocks and hence create an interface between size of input and encryption procedure. Considering this fact a scope for further experiments is found. In terms of throughput, Blow Fish gives a value greater than twice the throughput of the minimum value of AES. It is found that time required for encryption is fairly steady for input sizes variations within 310 KB size. RC6 and DES are observed to perform very close to each other in terms of time efficiency. In fact most of the curve's area appears to overlap in the graph. Again a need to compare time efficiency in terms of specific workloads is realised. Future work will focus on including complex parameters like nature and contents of input text and multiple whitespaces. Further study is desired to find relationship between the semantic nature of input and the encryption algorithms efficiency and speed. Though textual, image, audio as well as video data is ultimately stored as bits, data distribution as well as data density variations are related directly to the semantic nature of data. This relationship needs to be studied. Hence further scope is to relate the data density, data storage pattern and the semantic nature with the performance results of algorithms.

REFERENCES

[1] Anoop MS, "Public key Cryptography (Applications, Algorithms and Mathematical Explanations)"*IISc Library Archives* July, 2008

[2] Marshall D. Abrams, Harold J. Podell on Cryptography.*IISc Library Archives* December 2010.

[3] Hardjono, Security in wireless LANs and MANs, *Artech House Publishers*, 2005.

[4] Results of comparing tens of encryption algorithms using different settings Cryptoo++ benchmark. Retrieved on February 12, 2013, from http://www.eskimo.com/~wedai/benchmarks.html

[5] K. Nayak, D. S. L. Wei, "Software Implementation Strategies for Power Conscious Systems", *Mobile and Network Applications* – 6, 261-305, 2001

[6] William Stallings, "Cryptography and Network Security 4th Ed", *Prentice Hall 2005*, PP. 55-310.

[7] Daemen. J, and Rijmen. V, "Rijndael: The Advanced Encryption Standard" *Dr. Dobb's Journal*, March, 2001, PP. 137-139

[8] Coppersmith. D, "The Data Encryption Standard(DES) and its strength against attacks", *IBM Journal of Research and Development*, May 1994, pp 243-250.

[9] S. Hirani, "Energy Consumption of Encryption Schemes in Wireless Devices", *University of Pittsburgh,*. April 1998

[10] A. Nadeem, "A Performance comparison of data encryption algorithms", *IEEE Information and communication technologies*, pp. 84-89, 2006.

[11] P. Ruangachari, and P. Krishnamurthy, "Encryption and Power consumption in wireless LANs" *The third IEEE workshop on wireless LANs*, pp 148-152, Newtown, Massachusetts, Sep. 27-28, 2001.

[12] J. Kelaey, B. Schneier and D. Wagner, "Related Key Cryptanalysis of 3-way, Biham-DES, CAST, DES-X, New-DES, RC2 and TEA", *ICICS Proceedings",* November 1997

[13] Diaasalama, Abdul Kader, MohiyHadoud, "Studying the effect of most common encryption algorithms" *International Arab Journal of e-technology*, vol-2, no.1, January 2011.

[14] N. El-Fishawy, "Quality of Encryption Measurement of Bitmap Images with RC6, MRC6, and Rijndael Block Cipher Algorithms"*International Journal of Network Security*, November 2007, PP 240-250

[15] AamarNadeem et.al, "A Performance Comparison of Data Encryption Algorithms", *IEEE*, 2005

[16] Bruce Schneier, "Applied Cryptography" ,*John Wiley & Sons Inc*, 1996

[17] Behrouz A. Forouzan and Debdeep Mukhopadhyay, "Cryptography and Network Security 2th Ed", *Tata McGraw Hill*, 2012

[18] Results retrieved from http://searchsecurity.techtarget.com/definition/Data-Encryption-Standard on april 19, 2013

[19] retrieved from web source http://www.schneier.com/blowfish.html on March 20, 2012