I J C T A, 9(26) 2016, pp. 371-377

A Comparative Analysis on the Performance of Symmetric Block Ciphers

R.T. Anto Dafini[1] and J. John Raybin Jose[2]

ABSTRACT

Computer Security is an important aspect governing various aspects of data, authentication etc. From the early days till the present lots of algorithms are present to keep data safe using encryption techniques. This study is about the symmetric encryption models IDEA, AES, 3DES, Blowfish, RC6 etc. The Encryption time, Decryption time and the security strength are analyzed. The study performs a comparative analysis of the different techniques and presents the results for the user to decide on the selection of algorithm.

Keywords: Encryption, EET, Decryption, IDEA, Plain text, RSA, 3DES, AES.

1. INTRODUCTION

Symmetric-key algorithms use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys are identical and represent a shared secret between two or more parties for encryption and decryption. The major issue or drawback in symmetric key encryption when compared with the public-key encryption system. Normally stream ciphers encrypt bytes of a message one at a time only whereas block ciphers take blocks of bits and then encrypt them as a block (single) with appropriate padding in case of spaces to make the plaintext as a series of multiple blocks of same size. The study here is pertaining to the comparison of the DES, 3DES, RC6, Blowfish, IDEA and AES algorithms using plain text files of different sizes for both encryption and decryption.

2. LITERATURE OF REVIEW

1. Tool for Cryptographically Secure Statistical Analysis have proposed and implemented a suite of most used statistical analysis functions in the privacy-preserving setting including simple statistics, t-test, chi-squared test, Wilcoxon tests and linear regression and have given descriptionsof the privacy-preserving algorithms and benchmark results that show an order of magnitude improvement over previous work.
2. HOR ACE P. YU EN *et. al.* in their work proposed a simple but complete quantitativedescription of the information theoretic security of classical key distribution that is also applicable to thequantum situation current QKD proven security with that of conventional symmetrickey ciphers, and a list of objections and answers concerning some major points of this paper.
3. P. J. Escamilla-Ambrosio, M. Salinas-Rosales *et. al*. in proposed image compressive sensing is analyzed inorder to evaluate at what level it can be considered also as anencryption mechanism. This evaluation consists in performingsome security analyses and comparing results with thoseobtained with more traditional encryption algorithms as AES and Camellia.

[1] Research Scholar, Department of Computer Science, Bishop Heber College, Tiruchirapalli, India. *E-mail: rtantodafini2015@gmail.com*

[2] Associate Professor and Head, Department of Information Technology Bishop Heber College, Tiruchirapalli, India. *E-mail: raybinjosc@yahoo.com*

4. Ashokkumar C *et. al.* in employed a multi-threaded spy processand ensure that each time slice provided to the victim (running AES) is small enough so that it makes a very limited number oftable accesses. Where they designed implemented a suite of algorithmsto deduce the 128-bit AES key using as input the set of (unordered) cache line numbers captured by the spy threads in anaccess-driven cache-based side channel attack.
5. KURT PEHLÝVANOĐLU *et. al.* in WHT (Walsh-Hadamard Transform) that is used, oneof the transforms that used image processing techniques suchas attribute extraction on image files, text analysis, filtering,and compression. Image pixel values obtained at the end oftransformation encrypt with AES (Advanced EncryptionStandard) encryption algorithm.
6. Vandan Pendli in Algorithmanalyzed the effectiveness of the (Advanced Encryption Standard) AES algorithm by using Open MP API toreduce execution time and their process based on the results is confirmed that parallelcomputation reduced execution time as compared with sequentialcomputation.
7. Martin Abadi *et. al.* has proposed methods inenhancing the encryption and decryption of the plain text is bydouble encryption and double decryption which is proposed in where during the production of cipher text fromencryption the plain text is encrypted twice times and also decrypted twice. Since twice encryption and decryption takesmuch time this needs to be eradicated. The run time process inencrypting and decrypting the data takes much longer time, whichis huge drawback.
8. Chih-Chung Lu *et. al.* converting the plain text in to cipher text anddecrypting the data is by using the single core system. Hereonly one core is used irrespective of the size of the file which hasto be encrypted or decrypted. This method will work slowly if thedata file is big in size. It may work well for data files which aresmall but it is sure that it takes much longer time to encrypt anddecrypt the data for bigger files.

3. PROPOSED WORK

A. Data Encryption Standard Algorithm

DES is a block cipher, which encrypts the data in a block of 64 bits and produces the 64 bit cipher content where length of the key is 56 bits and at the start the key consists of 64 bits. The bit position 8, 16, 24, 32,40,48,56, 64 discarded from the key length the algorithm consists of the following steps:

In the first step, the starting 64-bits of plain text is made into a block and given to the [Initial Permutation] IP function. This acts on the text file producing two parts one part is a permuted block called Left Plain text and the (2^{nd}) second part is called Right Plain text. Further LPT an RPT undergo 16 rounds of encryption:

Key Transformation produces a 48-bit Sub-key from the 56-bit key.

Using the Expansion Permutation, the RPT is expended from 32 bits to 48 bits.

Next the 48-bit key is XORed with 48-bit RPT

The S-box substitution produces the 32-bit from 48-bit input.

P-Box permutes the 32 bits.

P-Box 32 bits XORed with 32 bits of LPT.

Swapping takes place. Next 32 bit XORed bits become RPT with LPT becoming RPT.

Similarly 15 more rounds are performed.

Finally after 16 rounds are completed the Final Permutation is performed [10], [17].

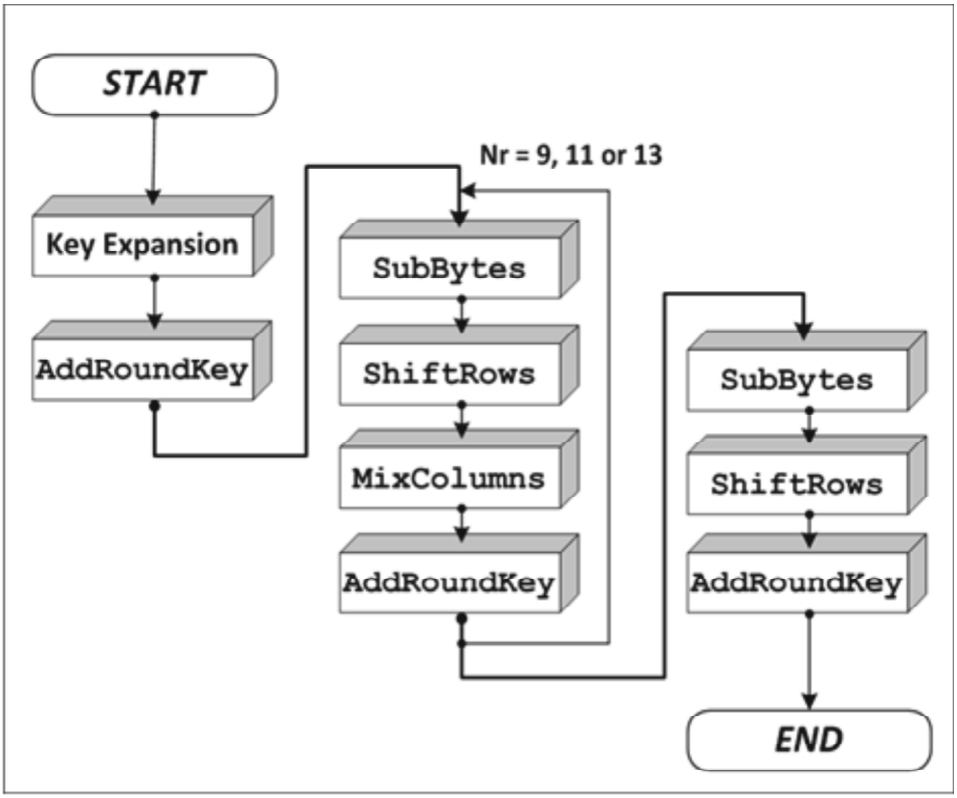

Figure 1: Architecture of AES Algorithm

B. Advanced Encryption Standard Algorithm

AES is a block cipher meaning that the number of bytes that it encrypts is fixed. AES can encrypt in blocks of 16 bytes at a time and no other block sizes are currently part of AES standard. In case encrypted bytes are larger than the AES specified block then it is executed concurrently within. Otherwise in case the given input plain text is less than 16 bytes it must be appropriately padded.

1st row is not shifted.

2nd row is shifted one (byte) position to the left.

3rd Third row is shifted two positions to the left.

4th row is shifted three positions to the left.

The result gives a fresh matrix consisting of the same 16 bytes but shifted with respect to each other. The other steps are as follows: Mix Columns, where each one column of four bytes is transformed into another new matrix which has 16 new bytes. After this Addroundkey is performed where 16 bytes of result matrix are XORed to 128 bits present in round key. The process of decryption in AES cipher text is similar to the encryption process but to be implementing in the reverse order. The process is described below.

- Add round key
- Mix columns
- Shift rows
- Byte substitution

C. TripleData Encryption Standard Algorithm

Triple DES is DES done three times and it is of two models one which uses three keys, and another which uses two keys. Initially the input plain text block *P* is encrypted with a key K1, then again encrypted with a second key *K*2, and for the third time encrypted with a key *K*3. Here *K*1, *K*2 and *K*3 are compulsorily different from each other. In order to decrypt the cipher text the operation $P = DK3\,(DK2\,(DK1(C)))$ should be performed in the reverse order. Here in Triple DES with two keys the algorithms work as follows:

Encrypt the plain text with key *K*1. Thus, we have *EK*1 (*p*).

Decrypt the output of step1 above with key *K*2. Thus, we have *DK*2 (*EK*1 (*P*)).

Finally, encrypt the output of above step again with a key *K*1 giving *EK*1 (*DK*2 (*EK*1 (*P*))) *D.*

D. IDEA (International Data Encryption Algorithm)

IDEA, unlike other block cipher algorithms is patented by Ascom a Swiss firm. However one cane use for free noncommercial purposes. IDEA is one of the best known as the block cipher algorithms. It uses a non-invertible hash function instead of the above described block ciphers and avoids lookup tables. The algorithm uses 52 sub keys with each 16 bits long. The algorithm Steps are as follows

The plaintext is split into *A*, *B*, *C*, and *D*, with the 52 sub keys being *K*(1) through *K*(52). Next the following is done: Multiply each part by the appropriate key *A* by *K*(1), next add *K*(2) to *B* in the third step add *K*(3) to *C*, further multiply *D* by *K*(4). Next step calculate *A* by xor in with *C* this is *E* and repeat *B* by xor ing with *D* this is *F*. Next step multiplying *E* with *K*(5), add this new value of *E* with *F* Finally multiply value of *F* with the *K*(6) and then add result *F* with *E*. Next replace both *A* and *C* by XORing *F* and change both *B* and *D* by XORing *E*. Finally exchange *B* with *C*. Repeat this step eight times using *K*(7) through *K*(12) for the 2nd time, up to *K*(43) through *K*(48) the eighth time.

E. Blowfish Algorithm

Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Since then 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. Blowfish is a fast block cipher, except when changing keys. Each new key requires pre-processing equivalent to encrypting about 4 kilobytes of text, which is very slow compared to other block ciphers. This prevents its use in certain applications, but is not a problem in others.

F. RC6

RC6 proper has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits. RC6 is very similar to RC5 in structure, using data-dependent rotations, and modular addition and XOR operations; in fact, RC6 could be viewed as interweaving two parallel RC5 encryption processes. However, RC6 does use an extra multiplication operation 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.

4. RESULTS AND DISCUSSION

The five different size text data files are given to the algorithms as input to check the performance of DES, 3DES, AES IDEA, RC6 and Blowfish. The experiment is performed on the machine [Intel® Pentium ® CPU G 630 @ 2.70 GHz, 2GB of RAM]. The operating system and system software used for these algorithms are Windows 7 in Java . The simulation results show the selected six encryption algorithms at different

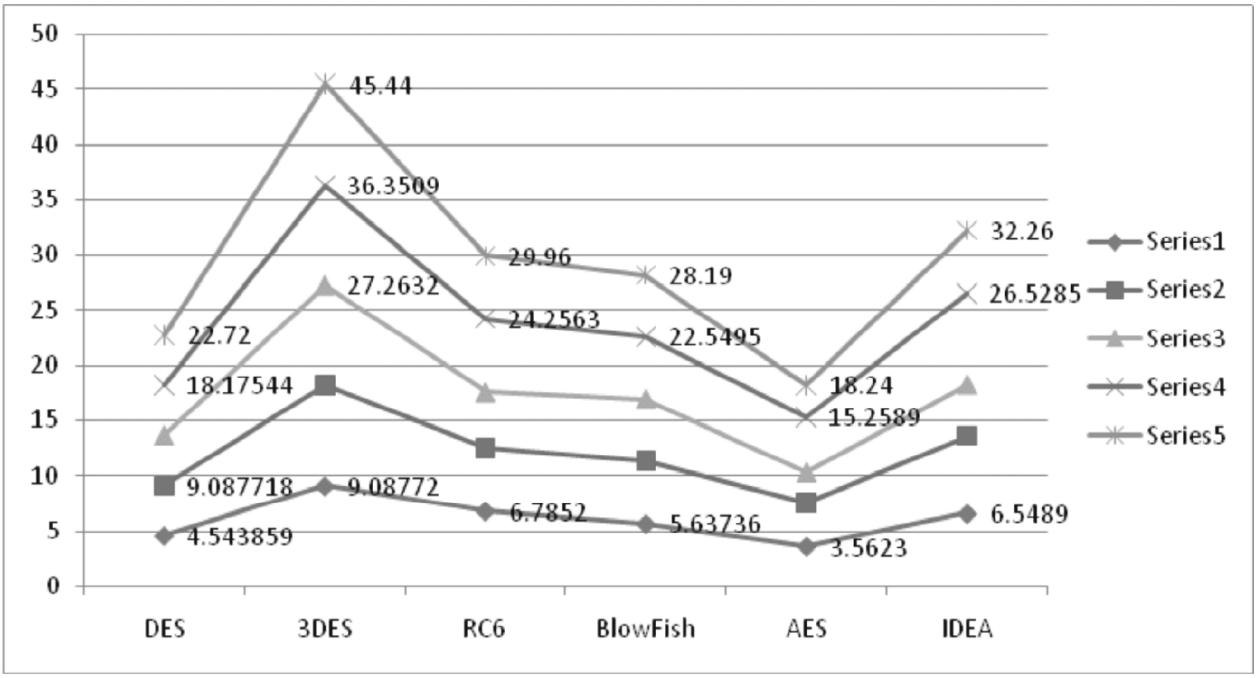

Figure 1: Comparison of Encryption algorithms

encoding method shows the results at base 64 encoding while the decryption results are of hexadecimal base encoding. One can notice that there is no significant difference at both encoding and decoding methods. The same files are encrypted by two methods; we can recognize that the two curves almost give the same results. The measured encryption time is gives the throughput of the encryption scheme which indicates the encryption speed. The throughput of the encryption scheme is calculated by dividing the total plaintext encryption done (in Megabytes) with the encrypted on the total encryption time done for each algorithm. When we see that the throughput value is increased, the power consumption of this encryption technique is decreased.

Table 1
Encryption Time

	Encryption Execution Time(Seconds)					
Input File Size (KB)	*DES*	*3DES*	*RC6*	*BlowFish*	*AES*	*IDEA*
15	4.543859	9.08772	6.7852	5.63736	3.5623	6.5489
30	9.087718	18.1754	12.5326	11.2747	7.52365	13.5689
45	13.63158	27.2632	17.5278	16.9121	10.25696	18.23569
60	18.17544	36.3509	24.2563	22.5495	15.2589	26.5285
75	22.7193	45.4386	29.9632	28.1868	18.2366	32.25689

Table 2
Decryption Time

	Decryption Execution Time(Seconds)					
Input File Size (KB)	*DES*	*3DES*	*RC6*	*BlowFish*	*AES*	*IDEA*
15	4.9982449	9.99649	7.46372	6.2011	3.91853	7.20379
30	9.9964898	19.993	13.7859	12.4022	8.27602	14.9258
45	14.994738	29.9895	19.2806	18.6033	11.2827	20.0593
60	19.992984	39.986	26.6819	24.8044	16.7848	29.1814
75	24.99123	49.9824	32.9595	31.0055	20.0603	35.4826

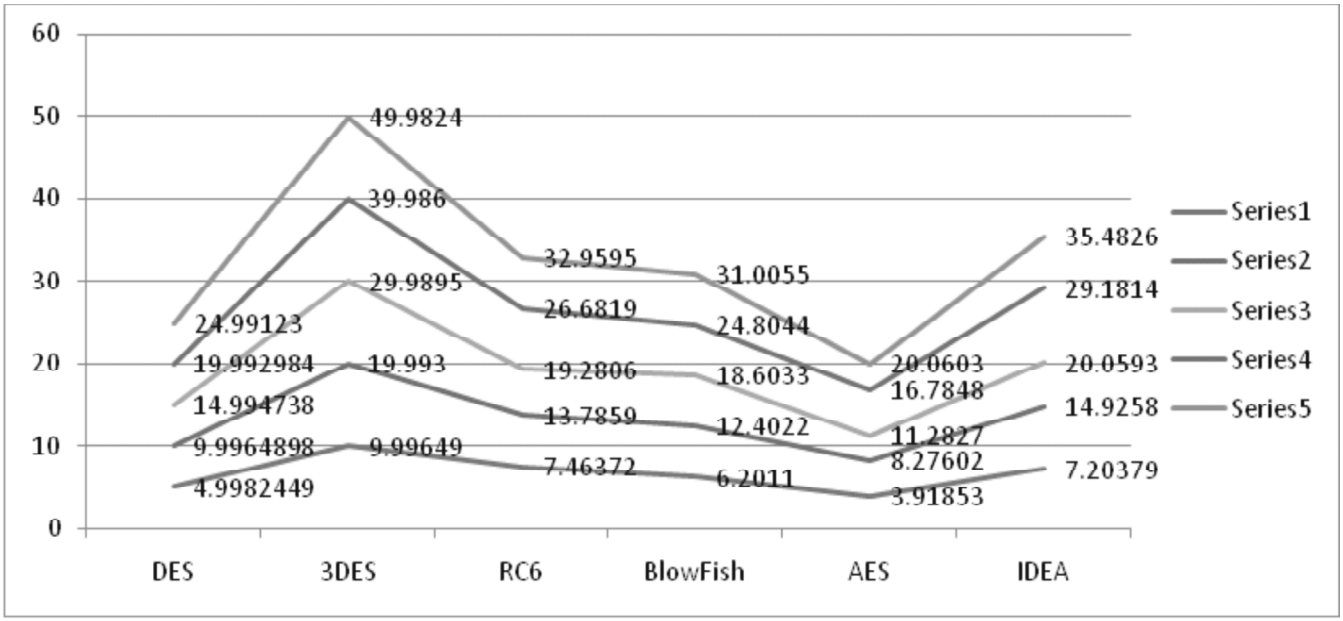

Figure 2: Comparison of Decryption algorithms

CONCLUSIONS

The encryption and decryption execution time consumed by various symmetric block ciphers such as AES,DES,3DES,RC6,IDEA,BLOWFISH algorithms are compared and analyzed.Among these algorithms the results obtained from AES indicates best performance.The encryption and decryption speed of 3DES algorithm is fast when compared to AES. The encryption execution time and decryption execution time consumed by AES algorithm are equal. The encryption execution time and decryption execution time consumed by RC6 and Blowfish algorithms are more or less same. The performance of DES is very good when compared to RC6. The throughput also explained that the encryption speed of DES is high when compared to RC6 or Blowfish algorithm. Decryption speed of DES algorithm is also less while comparing to RC6 algorithm. Thus in terms of safety and security and time AES stands out from the rest of the algorithms.

REFERENCES

[1] Dan Bogdanov, Liina Kamm, Sven Laur, and Ville Sokk "Rmind: a Tool for Cryptographically Secure Statistical Analysis" *IEEE Transactions on Dependable and Secure Computing* 2016.

[2] HOR ACE P. YU EN 2016 IEEE "Security of Quantum Key Distribution". *Defense Advanced Research Project Agency Cryptographic protocols.*

[3] P. J. Escamilla-Ambrosio, M. Salinas-Rosales, E. Aguirre-Anaya, R. Acosta-Bermejo "Image compressive sensing cryptographic analysis" *IEEE*2016.

[4] Ashokkumar C, Ravi Prakash Giri, Bernard Menezes"Highly Efficient Algorithms for AES Key Retrieval in Cache Access Attacks" *IEEE European Symposium on Security and Privacy*2016.

[5] Meltem KURT PEHLÝVANOĐLU1, Nevcihan "Encryption of Walsh Hadamard Transform Applied Images with the AES Encryption Algorithm" DURU11 Bilgisayar Mühendisliði Bölümü, Kocaeli Üniversitesi, Kocaeli, Türkiye.

[6] MartÍn Abadi, Roger Needham, "Prudent Engineering Practice for Cryptographic Protocols," *IEEE Transactions on Software Engineering*, **22(1)**, 6-15, January, 1996.

[7] Chih-Chung Lu; Shau-Yin Tseng, "Integrated design of AES (AdvancedEncryption Standard) encrypter and decrypter," *In Application-Specific Systems, Architectures and Processors*, 277-285, 2002.

[8] Diaasalama Abd Elminaam, Hatem Mohamad Abdual Kader, Mohly Mohamed Hadhoud, "Evaluation the Performance of Symmetric Encryption Algorithms", *international journal of network security* **10(3)**, 216-222, 2010.

[9] Diaasalama, Abdul kader, Mohiy Hadhoud, "Studying the Effect of Most Common Encryption Algorithms", *International Arab Journal of e- technology*, **2(1)**, 2011.

[10] Atul Kahte. "Cryptography and Network Security". *Tata Mcgraw Hill*, 2007.

[11] Shasi Mehlrotra seth, Rajan Mishra "Comparative Analysis of Encryption Algorithms For Data Communication", *IJCST*2(2), 2011.

[12] Wuling Ren. "A Hybrid Encryption Algorithm Based on DES and RSA in Bluetooth Communication". *Second International Conference on Modeling, Simulation and Visualization Methods (WMSVM),* 2010.

[13] Sung-Jo Han, Heang-Soo Oh, Jongan Park, "The improved Data Encryption Standard (DES) Algorithm", *Department of Electronic Engineering, Chosun University. South Korea*. 1996.

[14] "Optical Multiple-Image Encryption Using Three-Dimensional Space" in *IEEE Photonics Journal* April 2016.

[15] "Implementation of AES Algorithm to Overt FakeKeys against Counter Attacks"*International Conference on Computer Communication and Informatics (ICCCI-2016)*, Coimbatore, INDIA. 2016.

[16] Vandan Pendli "Improvising performance of Advanced EncryptionStandard"*Algorithm analyzed the effectiveness of the (Advanced Encryption Standard) AES algorithm IEEE* 2015.