iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** V **Month of publication:** May 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Feasibility Evaluation of Symmetric Key Encryption Techniques for Wireless Channel and Disk Storage

Md Asif Mushtaque[1], Mr. Khushal Singh[2]

School of Computing Science and Engineering, Galgotias University

Abstract*— Network security has become very important and challenging issue in this world. Cryptography is a technique that provides security for the sender and receiver to exchange their information over a network. There are basically two types of cryptography technique- Symmetric and Asymmetric. Symmetric key algorithm is also categorized into two parts Stream Cipher and Block Cipher. In this paper, we analyzed feasibility of Symmetric Key Block Cipher Encryption techniques for storage space with their advantages and disadvantages and also presented the possible attacks on these algorithms. The main objective of analyzing the feasibility for Storage space, because to increase the transmission speed over wireless channel it is necessary to reduce the size of the encrypted data.

Keywords*— Space complexity, Encryption Algorithm, AES, Serpent,, RC5, Rijndael, Symmetric Key, Cryptography, Private Key, Blowfish. Wireless sensor network.

I. INTRODUCTION

In this era of information technology, network security is very important to secure our data from unauthorized user; the information should be secure while transmitting from sender to receiver. Cryptography is a technique or method that provides security for data while we transfer our data over the network. Cryptography converts the information into another format that is not understood by anyone, unauthorized user can receives that format and try to convert in the original format but it is difficult for unauthorized user.

A. Some basic terms of cryptography:

Plain Text - *Original* message which is to be transferred.

Key - An alphanumeric value that is applied on the plaintext to convert its format.

Cipher Text- New format which cannot be understood, that is obtained after encryption of plain text.

Encryption - A process that converts plain text into ciphertext using a secret key. The encryption requires two main things an algorithm and a key.

Decryption- Decryption is a reverse process of encryption that converts ciphertext into the plaintext.

B. Processes of Cryptography:-

(i) The plaintext is encrypted using encryption algorithm and secret key; it is performed on the sender side.

(ii) Cipher text is obtained by applying the encryption algorithm on plain text.

(iii) On the receiver side encryption algorithm is applied in reverse process to obtain plaintext from ciphertext.

C. Classification of Cryptography

Cryptography algorithm can be classified into two categories- Symmetric and Asymmetric Key Encryption.

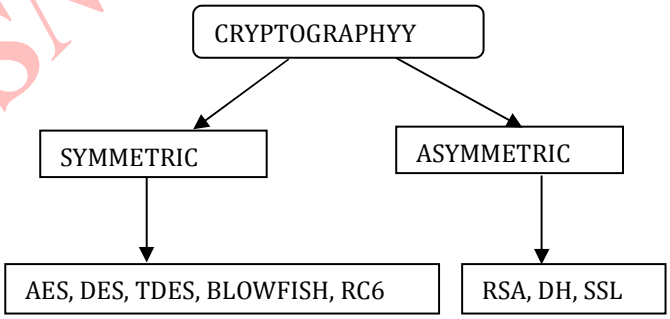


Figure 1: Classification of Cryptography

D. Types of Cryptography:

(i) Symmetric key Encryption

Symmetric key encryption is also called private key encryption. The same key is used to encrypt and decrypt the data. Private Key makes the encryption process fasted when it is used with a public key. But secret key cryptosystem suffers with the problem of exchanging the key. Different types of symmetric key algorithm has been developed, to apply any algorithm it is necessary to know about its advantages, disadvantages and their required parameters such as- security, encryption time, memory usage, flexibility and limitations. AES, DES, TDES and BlowFish, Modified Blowfish, RC5, RC6, are the example of symmetric key cryptography.

(ii) Asymmetric Key Encryption

Asymmetric key encryption is also known as public key encryption. This technique uses two different keys, one is used to encrypt message referred as the public key and another key is used to decrypt the message referred as a private key. RSA, Diffie-Hellman, Elliptic Curve Cryptography (ECC) are the example of asymmetric key encryption.

II. DESCRIPTION

In this section we present the description of symmetric key encryption algorithm like AES (Rijndael), Serpent, RC5, DES, TDES, BLOWFISH, Modified Blowfish etc. with their advantage and disadvantage.

A. AES (Rijndael)

Advanced Encryption Standard(AES) is the replacement of DES. It is a fixed size bit block of 128 bits and uses a variable key length of 128, 192 and 256 bits. AES performs a number of processing rounds which are based on size of the key. If the key size is 128 bits, it performs 10 rounds. If size is 192 bits it performs 12 rounds and if size is 256 bits, it performs 14 rounds [1].

The main advantage of AES is, it is fast and flexible. The space taken by ciphertext is more than plain text in AES.

In AES four different steps are followed. These are:

- Key_expansastion: In this step it derives the key from the cipher key by Rijndael key scheduling technique.
- Initial_round: In this step each byte is xored with key byte.
- Round: This step is divided into four sub steps.
  - (i) Sub_bytes:- each byte is replaced with its entry using S-boxes.
  - (ii) Shift_rows:- all rows shift in a cyclic way in an except first row.
    For example- in 2nd row each byte is shifted one step left side. In 3rd row each byte is shifted two steps left side and in last row each byte is shifted three steps.
  - (iii) Mix_column:- In this step each column is multiplied by some mathematical function.
  - (iv) Add_roundkey:- each byte of the state is xored with subkey.
- Final_round:- In this step all the steps of Round step are processed except mix_column step.

AES is based on substitution and permutation network and it is sufficient in software as well as hardware.

B. *DATA ENCRYPTION STANDARD (DES)*

DES was designed by IBM in (1976) and published National Institute of Standards and Technology with approval from National Security Agency (NSA). It uses 56-bit key length and 64 bit data block. In DES, key is stored as 8 bytes and 8th bit of each byte of the key is used for error detection at the time of key generation. DES is based on the Feistel Network so decryption is the same process as encryption it only takes the key in reverse order. Before the initial round it divides the 64 bit data block into two 32bit parts and goes

through 16 round Feistel Network with permutation process (initial and inverse permutation). Last bit of each byte is used for error detection due to this 56 key is used. Again, 64 bit data block is divided into two halves each of 32 bits and uses a uses a function, the Feistel Network function which defines f: $\{0, 1\}^{2n} \rightarrow \{0,1\}^{2n}$. Feistel networks are designed for the construction of secret cryptosystem. It was first proposed by Horst Feistel during his work on the cipher Lucifer at IBM. It is parameterized by the number of rounds $d \in N$ and the round functions $f1 \ldots\ldots ; fd : \{0, 1\}^n \rightarrow \{0, 1\}^n$ and it uses 12 to 16 rounds.

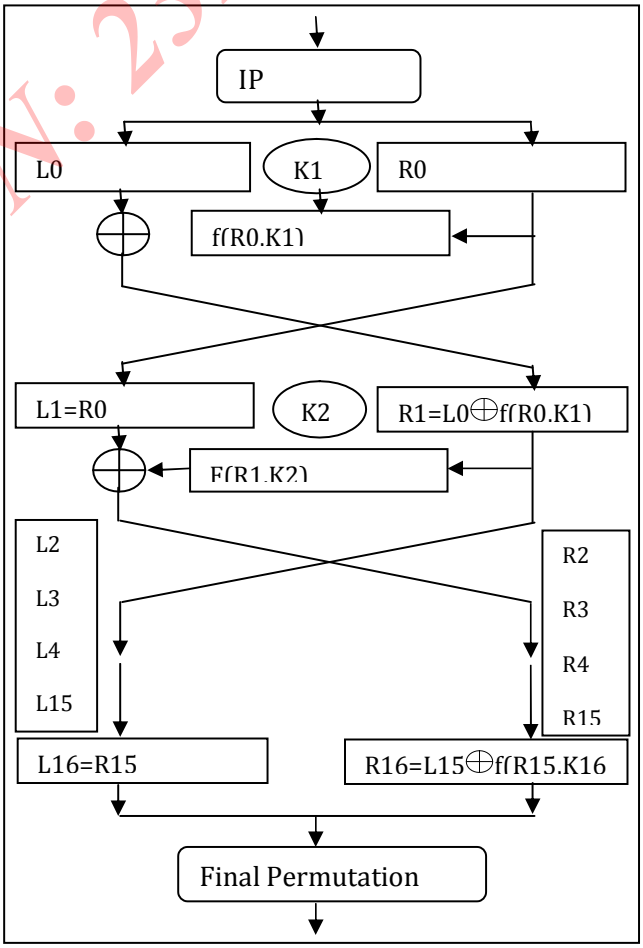


Figure 2. DES Encryption

ISSN: 2321-9653

Inputs are divided into two equal parts known as Left part and Right part [2, 4]. Then the R- function (round function) is performed on the right part and the obtained new right part is xored with left part. Then right part and left part are interchanged. Finally, inverse permutation is computed [4].

C. *TRIPLE DATA ENCRYPTION STANDARD (TDES):* DES was not secure because of its key length. So, TDES is derived from DES with three times encryption and decryption of a DES with three different keys [2]. It uses 64 bit text block as and uses (56*3) 168 bit key. As we know it is three times encryptions of DES i.e. only 112 bits key out of 168 is effective [2, 4]. TDES uses three steps in encryption and three steps in decryption of it means it uses DES three times. In a first step the information is encrypted using 1st key and in a second step, the cipher obtained from first step is decrypted with 2nd key and finally, in a third step, the obtain output of second step is again encrypted with 3rd key. The reverse process is followed in the decryption process of TDES, TDES takes 3 times more space than plaintext but it is secured than DES.

. Suppose, there are three different keys K1, K2 and K3 then

TDES Encryption performed as:

$CT= (Encptrypt)_{K3} ((Decrypt)_{K2} ((Encrypt)_{K1} (PT))).[4]$

TDES Decryption performed as:

$PT= (Decrypt)_{K1} (Encrypt)_{K2} ((Decrypt)_{K3} (CT)))$

Where, PT= PlainText, CT= Cipher Text, k_i ... key and i is iteration, and Encrypt and Decrypt are same process of DES encryption and DES decryption.

D. *BLOWFISH ENCRYPTION ALGORITHM*

Blowfish encryption is a symmetric block cipher encryption algorithm designed by Bruce Schneier in 1993. It is an alternative of the DES encryption algorithm. The same key is used by the sender to encrypt the information and by the recipient to decrypt the cipher text or information. It uses variable length key from 32 to 448 and 16 round Feistel cipher with key independent S-boxes. It has two parts key expansion and Encryption. In ken expansion part, 448 bit key is divided into several subkey and approximate total of these sub key is about 4168 bytes. It takes 64 bit block at a time and divide into two equal halves (A and B) each of 32 bits [7, 11], then iterates for 16 rounds.

- For i<=1 to 16
- Ai= Ai-1
- Bi=Bi-1 XOR F(Bi-1, Ki)

  Where Ki is the key in each round and F is a function which divides the 32 bit input into four quarters each of 8 bits. These quarters are used as input into S-boxes.
- F(XA)=((S1,a + S2,b mod232) XOR S3,c) + S4,d)

  Where a, b, c and d are four quarters.

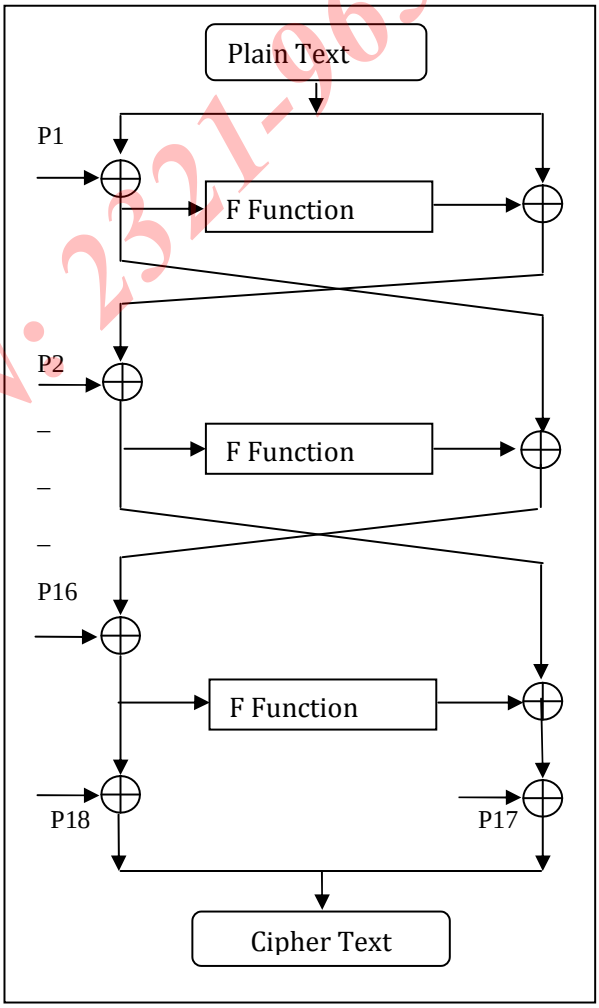


Figure 3. Blowfish Encryption

Blowfish is slow compared to other algorithms because each new key requires some pre-processing function to encrypt. Blowfish requires 3-4 times more memory size for cipher text as required by plain text.

E. *RC5*

RC5 is a block cipher algorithm created by Ronald Rivest (1994). It takes a variable size block 32, 64 or 128 bits, key size 0-2040 bits and it performs 0-255 processing rounds. In RC5, it is suggested to use a block size of 64bits, key size of 128 bits and number of rounds are 12 [12]. RC5 has several weak keys, if RC5 performs n round the probability of being a weak key for each key is $2^{\wedge -10n.}$ This is not a very big problem in RC5 it can be removed in the implementation. An attack designed by RC6 group can break RC5 algorithm up to 15 rounds. One of the most important key features of RC5 is that it performs data dependent rotation, it also performs addition and Xor operation and it uses two registers. RC5 is more accurately defined as RC5: W/R/B where 'W' is word size in bits, 'R' is the number of rounds and 'B' is the number of 8 bit byte in the key [12, 13].

The operations used in RC5 are as follows:

- A+B, integer addition modulo 2^W
- A⋘B, rotation of the W-bit word from A to the left by the amount given by the least significant log W bits of B.
- A⊕B, bitwise Xor of W-bit words

The key scheduling in RC5 is more complex because it expands the key by using binary expansions with a one way function. The 12 rounds of RC5 are observed for differential attack by using 2^44 selected plaintext.

F. *SERPENT ENCRYPTION ALGORITHM*

Serpent designed by Eli Biham, Ross Anderson and Lars Knudsen and it is the finalist of AES. It is based substitution and permutation structure; it uses a block size of 128 bits, a key length of variable size and performs 32 processing rounds on the block. It also uses the concept of s-boxes and consist 8 s-boxes [14]. The designer of Serpent guarantees that the 16 rounds of the serpent is enough to protect data against all types of attacks, but for future attack the algorithm can be extended to 32 processing rounds. Its security strength is stronger than other AES finalist. Initial half round of serpent provides better security than Triple-DES. The main disadvantages of serpent algorithm are that memory usage by this algorithm is very high and performance is low and very complex to implement with total 32 rounds, but due to its security level serpent algorithm stood at 2^{nd} position in the AES competition.

In serpent algorithm the R function known as round function performs some particular steps:

- Key_mixing using XOR
- 32 parallel applications on the same s-boxes
- Linear_ transformation

Linear_transformation performed in all rounds of the Serpent encryption algorithm except last one where another key_mixing operation is performed.

There are different types of attack tried to break the serpent algorithm but no any algorithm has been successfully broken all 32 rounds of the serpent.

In 2000, Kohno presented an attack that can break 6 of 32 rounds of the serpent. A 2001 attack by Nathan Keller, presented a cryptanalysis attack that can break 10 of 32 rounds. A 2011 presented by Hongjun Wu, Huaxiong Wang and Phong Ha Nguyen, they presented a linear cryptanalysis attack that breaks 11 of 32 rounds. The serpent encryption algorithm is very fast for hardware but it is little bit slow for software.

Serpent algorithm is flexible because the keys in serpent are padded to 256 bits; it consists of a '1' followed by '0'. The memory used by the serpent is almost equal to blowfish but the performance of blowfish algorithm is high for the same size of the file.

III. COMPARATIVE RESULT

Algorithm	Plaintext (Before Encryption)	Ciphertext	Plaintext (After Decryption)
AES (Rijndael)	130 KB	517 KB	130 KB
DES	130 KB	188 KB	130 KB
TDES	130 KB	360 KB	130 KB
Blowfish	130 KB	544 KB	130 KB
RC5	130 KB	517 KB	130 KB
Serpent	130 KB	480 KB	130 KB

Table1. Comparison Table based on memory usage

Factors	AES (Rijndael)	DES	TDES	Blowfish	RC5	Serpent
Cipher Text	Symmetric Block cipher	Symmetric	Symmetric	Symmetric	Symmetric	Symmetric
Key length	128,192,256	56	K1,k2,K3 168	32-448	0-2040	128,192,256
Block Size	128	64	64	64	32, 64 or 128	128
Rounds	10, 12, 14	16	48	16	1-255	32
Space Required for Cyphertext	3-4 times more space than plaintext	Almost 2 times more space required	2-3 times more space than plaintext	4 times more space than plaintext	3-4 times more space than plaintext	Almost 4 times more space required for small file
Possible Keys	2^128,2^192, 2^256	2^56	2^168	2^32 to 2^448	2^128,2^192 2^256	2^128,2^192,2^256
Effectiveness	In both s/w & h/w	slow	Slow specially in s/w	Efficient in s/w	Slow	Fast in H/W but slow in s/w
Speed	High	Low	Moderate	High	Moderate	low
Attacks	Side channel attacks	Brute Force Attacks	Theoretical possible	Not Yet	Differential Attacks	Not Yet

Table2. Comparison Table Based on Complete Architecture

Table2 shows the comparison between most common symmetric key encryption algorithms based on their architecture. From Table1 we analyze that only Blowfish and Serpent algorithm have been observed that these algorithms protect data from all kinds of attacks. In Table1 we compared the selected encryption algorithm on memory usage. So, from Table1 we analyze that DES requires less space among all these selected algorithms.

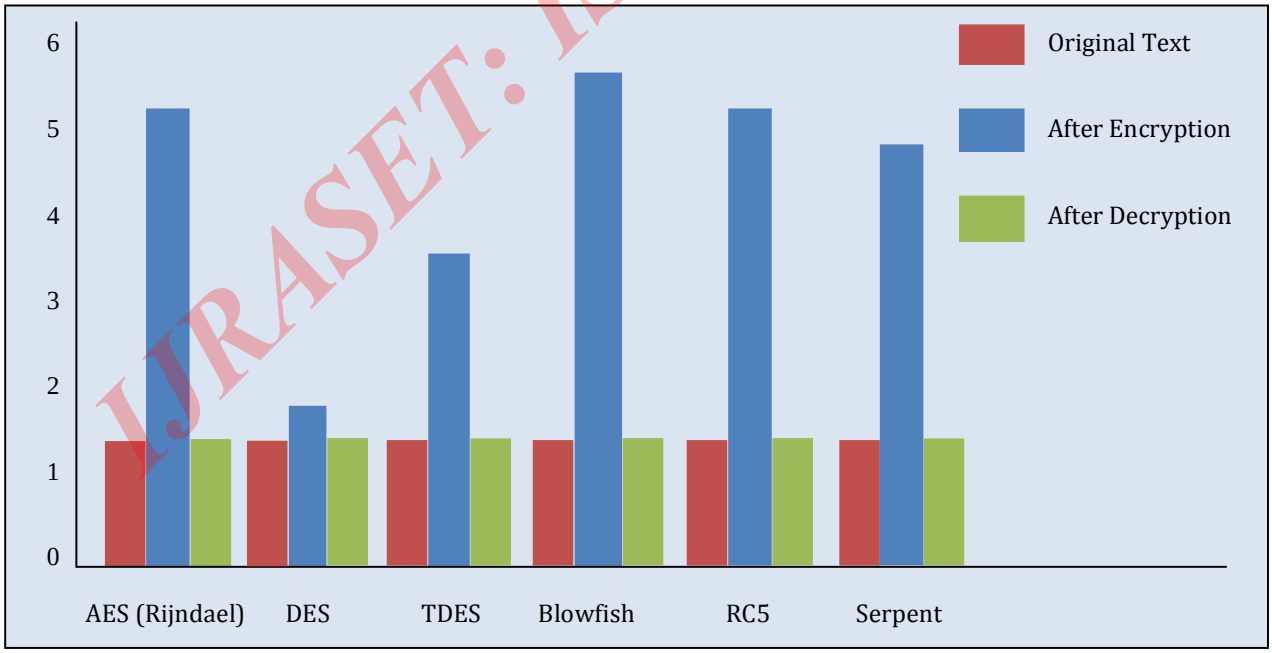


Figure 4 Graphical Representations of Algorithms on the Basis of Space Complexity

IV. Conclusion and Future scope

In this paper we discussed about the feasibility of various symmetric key block cipher encryption algorithm, Serpent, RC5, AES (Rijndael), DES, TDES, and Blowfish for wireless channel and disk storage. In this paper basically we discussed about how much memory space is taken by existing techniques. We analyzed that these entire algorithms require 3 to 4 times more memory space comparison to plaintext; according to [18], these are not more feasible in terms of storage space and for low bandwidth channel. Some algorithm takes extra space because of the variation in key length and block length. So, in the next paper we will propose our new Symmetric Key Encryption with minimum space complexity in comparison to these algorithms.

Acknowledgment

I would like to say thank to my Respected Guide Mr. Khushal Singh for his valuable guidance in all hard times and helping me to complete this work.

References

[1] Himani Agrawal and Monisha Sharma, "*Implementation and analysis of various symmetric cryptosystems*", Indian Journal of Science and Technology Vol. 3 No. 12, December 2010.

[2] Kruti R. Shah, Bhavika Gambhava, "*New Approach of Data Encryption Standard Algorithm*", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012.

[3] Prasun Ghosal, Malabika Biswas and Manish Biswas, "*A Compact FPGA Implementation of Triple DES Encryption System with IP Core Generation and On-Chip Verification*", Proceeding of the 2010 International Conference on Industrial Engineering and Operation Management, Dhaka, Bangladesh, January 9-10-2010.

[4] Shashi Mehrotra Seth and Rajan Mishra, "*Comparative Analysis of Encryption Algorithms for Data Communication*", ISSN: 2229-4333, IJCST Vol. 2, Issue 2, June 2011.

[5] Russell K. Meyers and Ahmed H. Desoky," *An Implementation of the Blowfish Cryptosystem*", IEEE- 978-1 -4244-3555-5/08, 2008.

[6] Rasheed Mokhtar Ahmed, Adel Zaghlul Mahmoud, "*An Implementation of High Security and High Throughput Triple Blowfish Cryptography Algorithm*", IJRRSAP Vol. 2, No. 1, ISSN: 2046-617X, March 2012.

[7] M. Anand Kumar and Dr. S. Karthikeyan, "*Investigating the Efficiency of Blowfish and Rejindael (AES) Algorithms*", I. J. Computer Network and Information Security, 2012, 2, 22-28 in MECS, March 2012.

[8] S. M. Dehnavi, M. R. Mirzaee Shamsabad, A. Mahmoodi Rishakani and Einollah Pasha, "*Generalization of Statistical Criteria for Sboxes*", IEEE, 978-1-4673-2386-4/12, May2012.

[9] Ralf Kusters and Tomasz Trundrerung, "*Reducing Protocol Analysis with XOR to the XOR-free Case in the Horn Theory Based Approach*", ACM 978-1-59593-810, October, 2008, Alexandria, Virginia, USA.

[10] Majdi Al-qdah and Lin Yi Hui, "*Simple Encryption/Decryption Application*", International Journal of Computer Science and Security (IJCSS-4), Volume (1), December 2011.

[11] Gil-Ho kim, Jong-Nam Kim "*An improved RC6 algorithm with the same structure of encryption and decryption*", IEEE, ISBN 978-89-5519-139-4, Feb-2009

[12] T.Gunasundari and Dr. K.Elangovan "*A Comparative Survey on Symmetric Key Encryption Algorithms*", International Journal of Computer Science and Mobile Applications, Vol.2 Issue. 2, February- 2014.

[13] G. Muthukumar and Dr. E. George Dharma Prakash Raj, "*A Comparative Analysis on Symmetric Key Encryption Algorithms*", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3, Issue 2, February 2014.

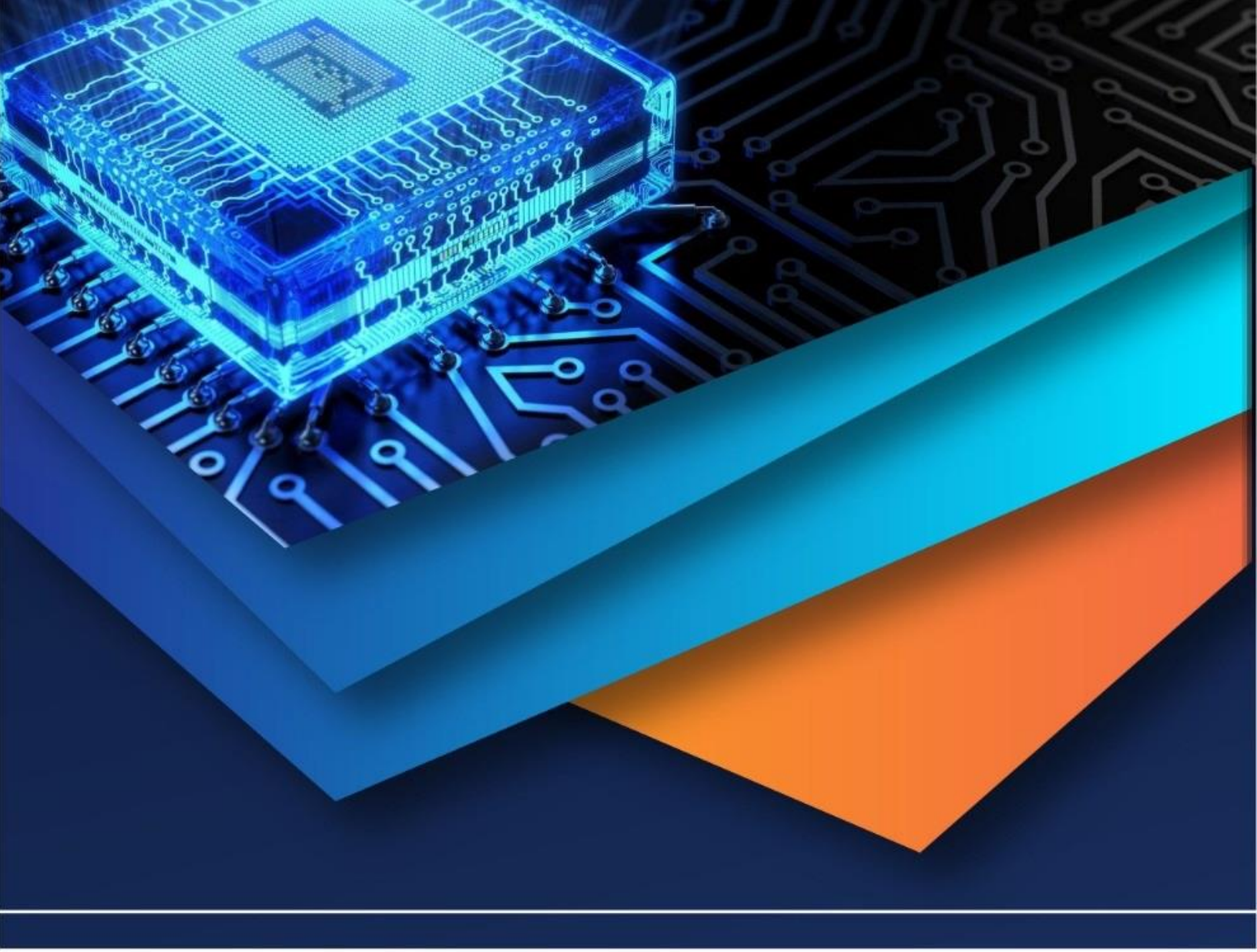
[14] Mansoor Ebrahim, Mansoor Ebrahim and Mansoor Ebrahim," *Symmetric Algorithm Survey: A Comparative Analysis*", International Journal of Computer Applications (0975 – 8887) Volume 61– No.20, January 2013.

[15] Tingyuan Nie, Yansheng Li and Chuanwang Song," *Performance Evaluation for CAST and RC5 Encryption Algorithms*", International Conference on Computing, Control and Industrial Engineering, IEEE, 2010.

Akash Kumar Mandal, Chandra Parakash and Mrs. Archana Tiwari," *Performance Evaluation of Cryptographic Algorithms: DES and AES*", 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science.

[16] A.Ramesh and Dr.A.Suruliandi, "*Performance Analysis of Encryption Algorithms for Information Security*", International Conference on Circuits, Power and Computing Technologies [ICCPCT-2013], IEEE, 2013.

[17] http://en.wikipedia.org/wiki/Disk_encryption_theory accessed on 24th April 2014.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY