An Axiomatic Approach for Secure Data Transmission

Srishti Agarwal
Assistant Professor
Cs & It Dept.
Mit, Moradabad
U.P., India

Anurag Malik
Associate Professor
Cs & It Dept.
Mit, Moradabad
U.P., India

ABSTRACT
Cryptography is the strongest link in the chain of security since it plays an integral role in secure communication. Symmetric key cryptography is the oldest form of cryptology in which identical keys are used by the sender and receiver to share the confidential data. This research paper is about analyzing and designing of a symmetric key cryptographic technique which provides security at enhanced level and offer defense against non-traditional security threats. The algorithm "An Axiomatic Approach for Secure Data Transmission" is compared with the algorithm "Advanced Encryption Standards", analytical results are also shown to prove the better outcomes.

Keywords
Symmetric Key Cryptography, Cipher text, Block Cipher Encryption/Decryption, Minimized CPU Utilization, Increased Key Complexity, Minimized Memory Utilization, Advanced Encryption Standards.

1. INTRODUCTION
Modern society and modern economies rely on Internet for data transferring, communication, transportation, finance and much more. But hackers are all around over internet to harm people by gathering or even alternating their personal information. Attacks on data travelling on network can threat the personal and economical security of organizations and people. So it requires developing some software to provide security as much as possible. Developing secure software systems for communication correctly are difficult and error-prone. [8] Cryptographic techniques are required across variety of platforms in different areas these days. If data is not encrypted then it can suffer from many problems and these can be accidental altering of data, transmission error or even hard disk crash. Thus data encryption is very important and cryptography has wide scope in many fields [10]. Security of any encryption algorithm depends greatly on the key used for encryption/decryption.

Private Key cryptography where same key is used for encryption and decryption face a great challenge of sharing the key among sender and receiver [3]. Thus one important aspect to consider during designing of cryptographic algorithm is security. Where in public key cryptography there is no requirement for key sharing, they suffer with a problem that they are very much CPU intensive. So, one thing to have impact focus along with data security is CPU utilization. Also, it is known that more memory utilization decreases the system's performance, so it causes decrease in memory utilization and is also a concern to be focused on [6]. Thus goal of An Axiomatic Approach for Secure Data Transmission Algorithm is to maximize the security and along with that minimize CPU utilization and Memory Utilization.

2. BASIC CONCEPT AND WORKING BEHIND "AN AXIOMATIC APPROACH FOR SECURE DATA TRANSMISSION"
As we know that, the upsurge of information technology and increase in the dependence on electronic devices, main motive of cryptography has been expanded from coding/decoding to authorization and security of the information [5]. The vast digital data that is stored and processed in large computer bases that is transmitted through complex communication networks is susceptible to unauthorized interception and interpretation and hence, needs to be protected through encrypted remote access or passwords. To prevent data theft, we are in need of cryptographic techniques with stronger codes to provide more and more security [1].

The main focus of this algorithm is "Security" along with the efforts to reduce CPU utilization and memory Utilization and thus enhancing the working speed. An Axiomatic Approach for Secure Data Transmission is a Secret Key Cryptographic Technique using block cipher encryption method, in which a key of 512 bits in binary (or 128 bits in hexadecimal) format is generated on the sender side by taking some inputs from the user and applying some mathematical concepts on those values. As there is a saying about cryptography that "Engineering Security meets with Mathematics" [12], thus in this algorithm different and complex mathematical concepts are embedded with the motive of enhancing the key complexity and thus security of the algorithm. This key is used to encrypt/decrypt data in blocks of 512 bits each time.

The best security aspects and features added to the algorithm are as follows:

- For each and every transaction it generates a new key. Key change does not depend on the time it has been used for, rather after every transaction whether it's of 1 sec or of 10 sec, new key will come in process.
- For same inputs each and every time a different key will get generated. Since the algorithm is generating 512 bits key and each time it will be different, so it's very difficult to understand the key pattern by an unknown person and thus the algorithm is very much secure.
- It is not only encrypting alphabets or numbers, rather this algorithm support encryption of special symbols too…

2.1 Basic Working of the Algorithm
The algorithm works as follows:
Step 1: User has to input a large value preferable prime, which is divided into two parts and these two values are checked that whether they are also prime or not. If they are prime then the process continues, otherwise prime values

closest to these values are found and are used for further processing [7]. This is just for the reason we know that prime values are hard to crack since these cannot be factorize.

Step 2: One more value is required for further working and it is also divided into two parts. And these two values should not be equal.

Step 3: These all values are then used to generate a matrix which can be of variable size each time. But it should be of minimum 3*3 just for the security purpose.

Step 4: On that matrix row and column permutations are applied so as to shuffle the values which increases security, because more shuffling creates more options for different keys in this algorithm.

Step 5: Then each cell of both the matrix is converted to 8 bit binary number. And all corresponding cells of both matrices are XORed to get a final matrix.

Step 6: Then this matrix is used to generate a large key of 512 bits in binary. From this matrix different cell values are selected randomly and concatenated together to form a key.

2.2 Key Generation Process

On performing permutations on matrix, a matrix is shuffled in a manner so as to get each value shuffled row wise and then column wise. Thus after shuffling, different matrix combinations can be generated. Thus for a single input it can generate $((column)^{row} * (row)^{column})$ different options.

On getting a final matrix after the permutation and XOR function, values from matrix are fetched on random basis and concatenated together to generate a key of 512 bits. Since each cell is comprised of 8 bit binary number, so it just require to randomly taking any cell and concatenate all of them until and unless a key of 512 bits is prepared.

If the matrix is of pc*qc size, then each time when fetching the value it would have pc*qc different options and there would be total 64 combinations for 512 bit key. So complexity of key fetching becomes (64*pc*qc)

Thus finally the Complexity of Key Generation is $(((column)^{row} * (row)^{column})*((64 * row * column))$

2.3 Encryption/Decryption Process

The process of encoding a message or manipulating the data based on password (or key) for security purpose, so that it cannot be easily understood by unauthorized people is known as encryption. This non readable form of data is known as cipher text [9]. The reverse process where the encrypted data is converted back to its original form is known as decryption [2].

This algorithm follows the concept of block cipher encryption/ decryption. Thus break the whole cipher text into blocks each of 512 bits since the key size is also 512 bits.

Perform encryption in a way that, XOR the first block of plain text with the key to form a cipher text. Let PT1 XOR with key provide CT1. Then second block will first get XOR with the previous cipher text which is taken in reverse order and then with the key to provide a new cipher text. And the process will get continued until whole plain text is converted to the cipher text.

During decryption, first block of cipher text will be decrypted by the key and the next blocks will be decrypted by previous cipher text in reverse order and key both as we used in case of encryption.

Table 2.1: Encryption and Decryption Process of "An Axiomatic Approach for Secure Data Transmission"

Encryption:	Decryption:
PT1 XOR K = CT1	CT1 XOR k = PT1
PT2 XOR CT1 (reverse order) XOR K = CT2	CT2 XOR CT1 (reverse order) XOR K = PT2
PT3 XOR CT2 (reverse order) XOR K = CT3	CT3 XOR CT2 (reverse order) XOR K = PT3
And so on….	And so on….

Where PT is for Plain Text and CT is for Cipher Text.

2.4 Complexity of "An Axiomatic Approach for Secure Data Transmission" Algorithm

In algorithm matrices are considered which depends on factors of two different values. These factors are set to minimum of 3*3 since a matrix less than this size can be easily cracked and for maximum it can be any value based on a value entered by user. Let us consider matrix size to be M*N. In programming nested loops are there, while performing shuffling or any other permutation on these matrices in which outer loop works for 0 to M and inner loop for 0 to N. Thus the complexity of algorithm is **O(M*N).**

3. ANALYTICAL RESULTS

It have been understood that key/data security and minimized utilization of resources are the most important topic to be focused on and need more and more attention and research to improve the performance of cryptographic algorithms.

The main focuses considered during the designing of "An Axiomatic Approach for Secure Data Transmission" are:

- A new key is generated for each run of the code. Even for same inputs entered by the user key will be different.
- Minimization of CPU Utilization and Memory Utilization.
- Key with large size since key with small size can easily be cracked.

Various inputs are given to the algorithm to measure CPU utilization. And this utilization is also measured by running Advanced Encryption Standard (AES) algorithm. Comparisons between two algorithms are measured based on CPU Utilization and Memory Utilization.

In "An Axiomatic Approach for Secure Data Transmission" CPU Utilization is calculated on two basis, first is giving different inputs on different run of the program where graph is calculated between inputs given and obtained CPU utilization refer to Figure 3.1 and second method where CPU utilization is calculated for one full run of a program where utilization measurement starts when the program is run and end when final output comes, here graph is plotted between time and CPU Utilization as in Figure 3.3 along with its snapshots in Figure 3.2. This second method is also adopted for CPU Utilization of AES algorithm [4] shown in Figure 3.5 and its snapshot in shown in Figure 3.4. From these figures, it can be easily understood that CPU Utilization of An Axiomatic Approach for Secure Data Transmission is reduced to a great extent in comparison to Advanced Encryption Standards. One more method is used for comparison between two algorithms where memory utilization is checked and a combined graph is plotted for their comparison shown in Figure 3.6. "An Axiomatic Approach for Secure Data Transmission" is better

than "Advanced Encryption Standard" in this manner also, where less memory is utilized.

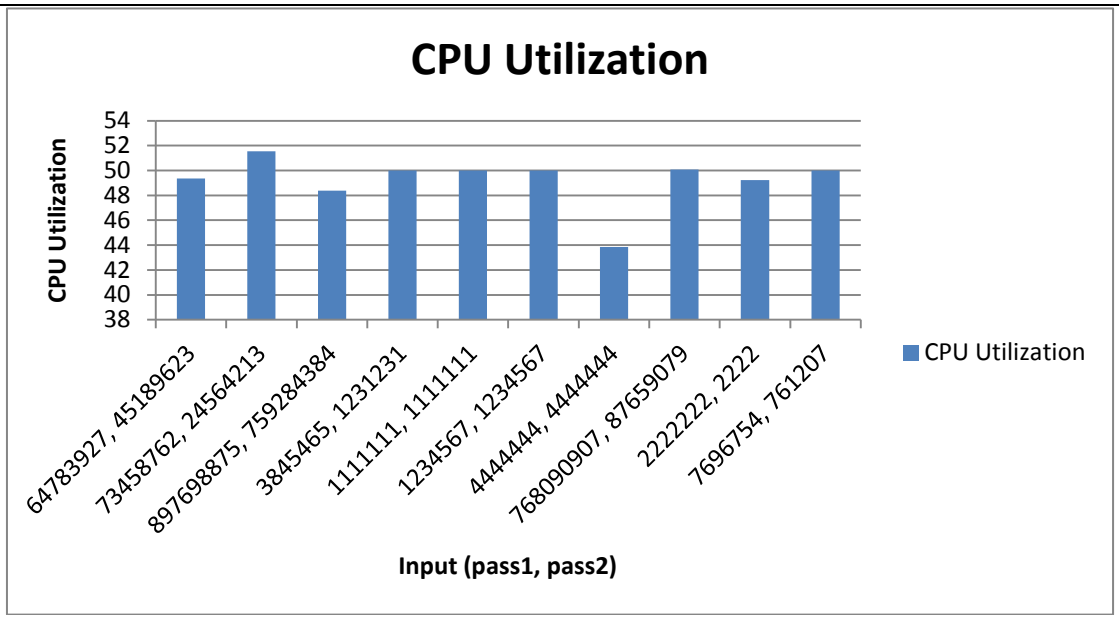

Figure 3.1: Graph Showing Relation between Different Inputs Given to "An Axiomatic Approach for Secure Data Transmission" Algorithm and Total CPU Utilization for that Input

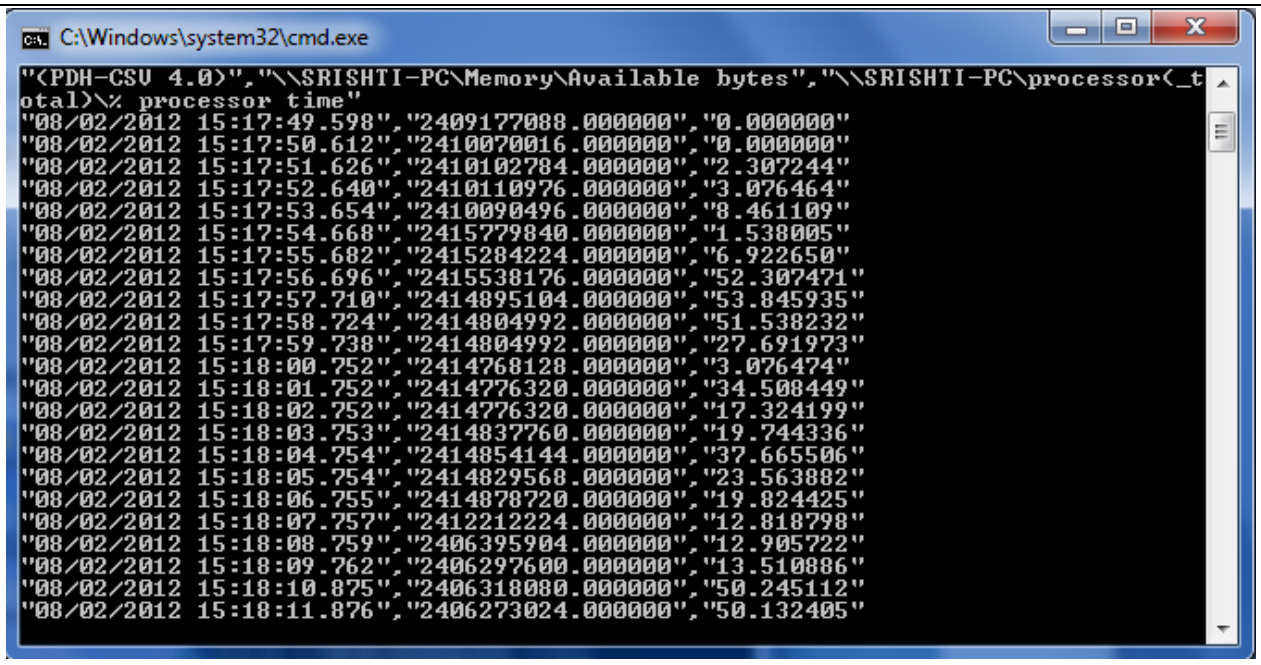

Figure 3.2: Snapshot of CPU Utilization During One Full Run of "An Axiomatic Approach for Secure Transmission" Program

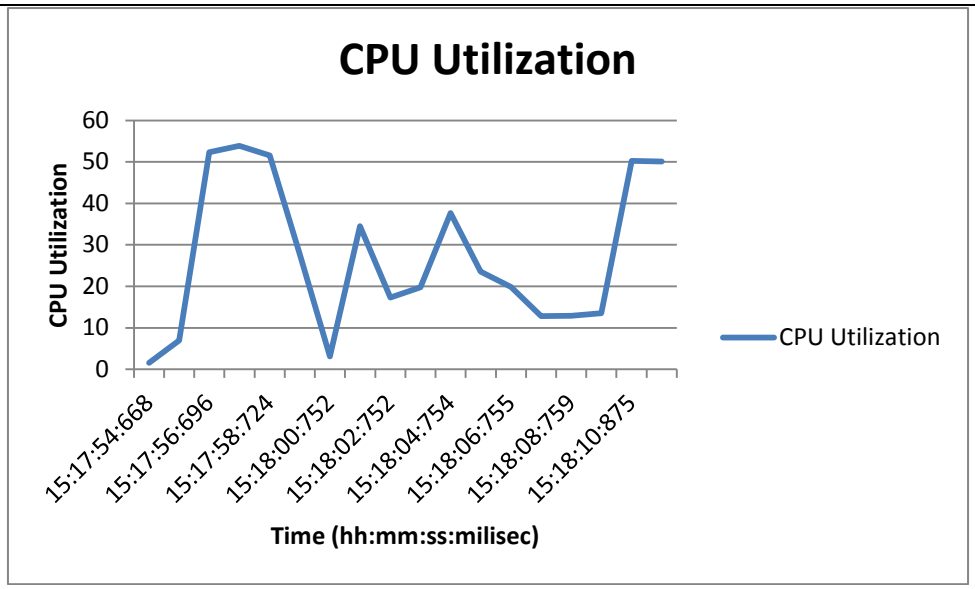

Figure 3.3: Graph Showing CPU Utilization for One Full Run of "An Axiomatic Approach for Secure Transmission" Program

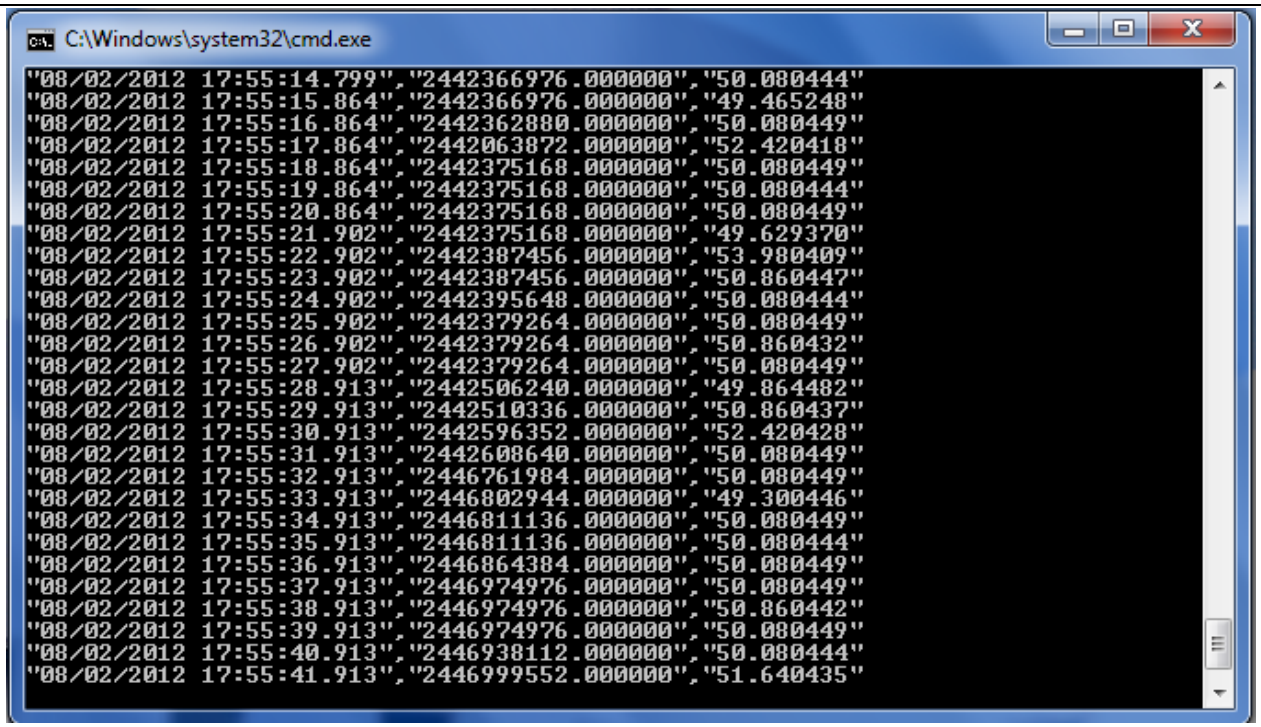

Figure 3.4: Snapshot of CPU Utilization during One Full Run of "Advanced Encryption Standard" Program

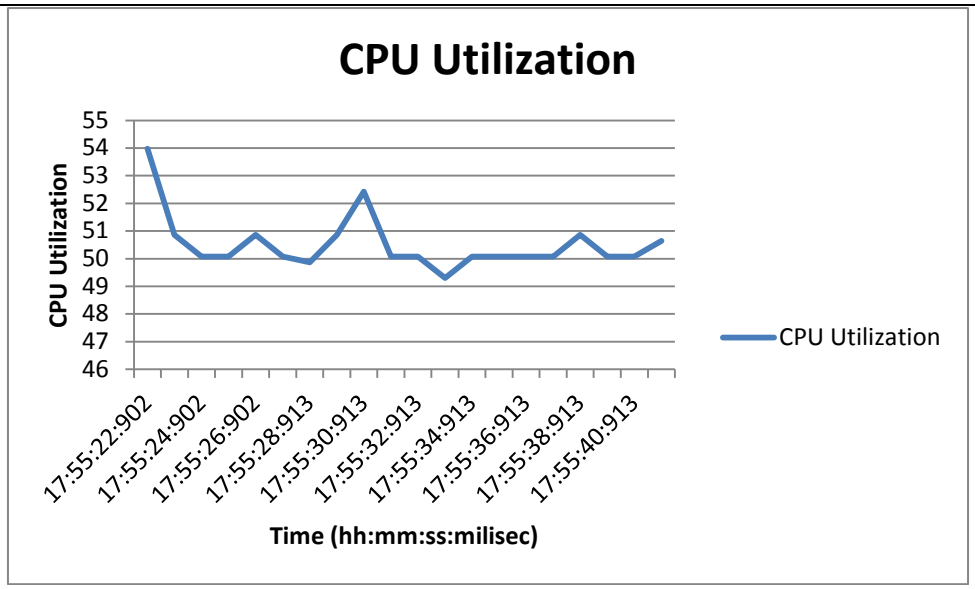

Figure 3.5: Graph Showing CPU Utilization for One Full Run of "Advanced Encryption Standard" Program

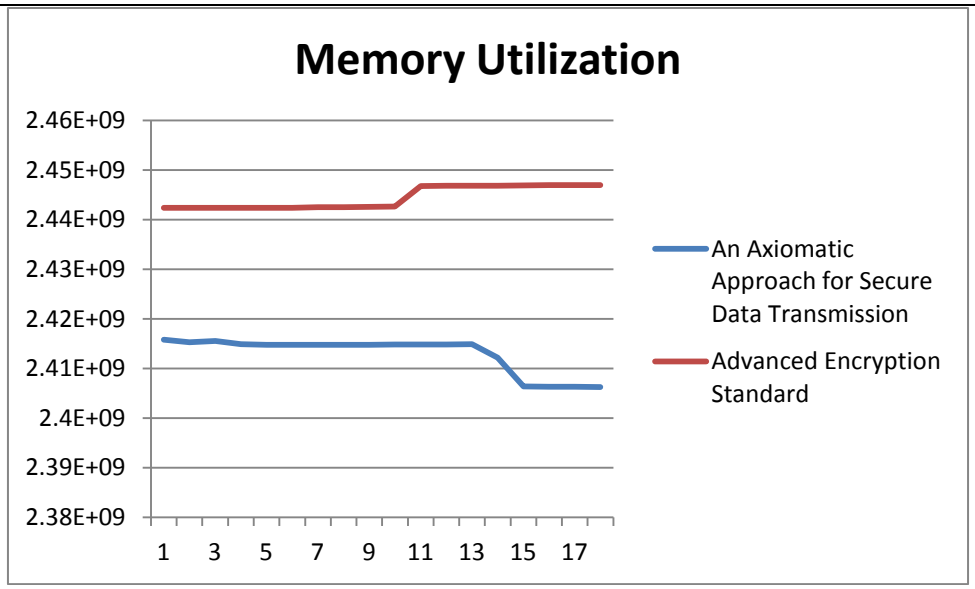

Figure 3.6: Comparative Analysis of Memory Utilization between "An Axiomatic Approach for Secure Data Transmission" and "Advanced Encryption Standard"

4. RESULT COMPARISON

A comparison is conducted between the results of "An Axiomatic Approach for Secure Data Transmission" and "Advanced Encryption Standards" based on CPU and Memory Utilization. From all the Figures shown from 3.1 to 3.6 it can be said that "An Axiomatic Approach for Secure Data Transmission" algorithm is giving better results in comparison with "Advanced Encryption Standard" algorithm in both manner whether it's about CPU Utilization or Memory Utilization. So with this algorithm I can say that I have succeeded to reduce both CPU as well as Memory Utilization to a measurable extent.

5. CONCLUSION

With all the work shown and graphical representation of comparative analysis in previous chapter, I can conclude that I have succeeded in reducing CPU Utilization to a large extent which was the main motive of this research work. And along with this my algorithm is also reducing memory Utilization thus it can be said that "An Axiomatic Approach for Secure Data Transmission" is better in comparison with "Advanced Encryption Standard" in both ways i.e. CPU Utilization and Memory Utilization both are reduced in this algorithm.

Along with all this I cannot ignore the fact of key complexity that it should be large enough and complex so as not easily

breakable. And as shown in section 2.2 key complexity of this algorithm is very large and cannot be easily cracked. Since it is producing different key values each time it is run. One more point to remember is that as we all know with the increasing use of network, algorithm with more and more security is required always and it will be very favorable for an encryption algorithm if its key size is variable. Since "An Axiomatic Approach for Secure Data Transmission" is an algorithm which is generating key from a matrix taking its values randomly until key of 512 bits is not formed, it can be said that any time if algorithm requires increased key value it can be easily done.

6. FUTURE DIRECTION

We are in an era where we require extremely strong symmetric key cipher and should have a technique by which every communication becomes secure without any overhead of the user [11]. "An Axiomatic Approach for Secure Data Transmission" is an algorithm with minimized CPU Utilization along with less memory utilization and enhanced security due to key complexity thus it can be in-built in an Operating System. So that without any background knowledge user's each communication become secure and he can securely communicate his personal information without fear of data loss and hack.

7. REFERENCES

[1] Ayushi "A symmetric Key Cryptography" Published in International Journal of Computer Applications (0975-8887), Vol. 1, No. 15, pp 1-4

[2] H. Lee Kwang "Basic Encryption and Decryption"

[3] Himani Agrawal and Monisha Sharma "Implementation and analysis of various symmetric cryptosystems", Indian Journal of Science and Technology Vol. 3 No. 12 (Dec 2010) ISSN: 0974- 6846, pp. 1173-1176

[4] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[5] http://williamstallings.com/Crypto/Crypto4e.html

[6] http://www.garykessler.net/library/crypto.html

[7]http://www.go4expert.com/forums/showthread.php?t=19704

[8] http://www-jj.cs.tu-dortmund.de/secse/pages/research/motivation_en.shtml

[9] http://www.topbits.com/plaintext-and-ciphertext.html

[10]http://www.ukessays.co.uk/essays/engineering/background-and-research-motivation.php

[11]mbandukda.files.wordpress.com/2008/03/cryptosystems.doc

[12] www.cl.cam.ac.uk/~rja14/Papers/SE-05.pdf