A Non-Feistel Symmetric Key Encryption Algorithm

Raza Ali and Dr. Muhammad Ali

Abstract— **Network security is the primary element to secure a communication network, and if an intruder still able to break this security element, the other and an important element is data encryption which hides the intelligence of data so that other than authorized entities cannot understand the encrypted data. This paper proposed a new shared key, block cipher encryption/decryption algorithm. The algorithm is designed keeping the features of non fiestel cipher and counter mode of block cipher is used in the algorithm. The encryption process starts by entering a key and counter value which is 128-bit long. From the 128-bit of key, decimal value is calculated and added between two counter values to make counter value 128-bit. To encrypt data, 128-bit key is used along with counter value. This new algorithm encrypts 128-bit plaintext block and produces 128-bit cipher text block using several encryption elements including matrix transposition, adding average value, key based bitwise transposition operation, and byte substitution. The algorithm is designed for secure communication applications. For the analysis of proposed algorithm different cryptanalysis techniques such as frequency attack, avalanche effect, throughput time and brute force attack are used, and the design is useful in terms of privacy, data confidentiality. The unique combination of encryption elements obscures the relationship between plaintext and cipher text.**

Index Terms — **Encryption, Decryption, Key, Counter, Symmetric key cryptography, non-fiestel cipher.**

I. INTRODUCTION

Data security is considered as a major issue in today's world of communication. Information is considered as an asset between two sharing parities, which requires that their information will be kept unavailable, inaccessible, and strange to rest of the world. Information is transformed from plain text to cipher text and vice versa [1]. This transformation can be done by either of the two ways; symmetric key cryptography or asymmetric key cryptography [2]. For encryption, many symmetric and asymmetric key based, cryptographic algorithms have been designed to provide secure communication such as DES (feistel cipher, symmetric), AES (non-feistel, symmetric) [3], RSA (asymmetric) [4] and ECC (asymmetric) [5]. These algorithms are used to ensure the three distinct goals of security which are confidentiality, integrity, and availability [6]. Cryptographic algorithms are broadly divided into two categories, Symmetric key cryptography and asymmetric key cryptography.

Asymmetric key cryptography also known as public key cryptography uses two keys, public key for encryption and private key for decryption. Public key cryptography is based on the creation of mathematical puzzles that are difficult to solve without certain knowledge about how they are created [7]. Symmetric key cryptography uses a single key called shared key for encryption and decryption. Key needs to be shared over a secure channel. Symmetric key algorithms are faster than asymmetric key algorithms as they require more processing power, [8] but are more secure than symmetric algorithms. Symmetric key encryption algorithms are still widely used as powerful techniques in insecure communication channel [9]. For block cipher, several symmetric key cryptography algorithms are proposed with different encryption techniques to make data secure, based on algorithm computation [10], to have high security with less consumption of resources and improved the performance speed [11], low computational power and encryption/decryption for input of any length and key size in constant time using property of circle, and circle-centered angle [12], dynamic substitution rule, used in substitution step to achieve lower encryption time and high throughput [13], uses XOR operation for the chaining process to achieve fast and secure encryption [14], and hybrid encryption technique that generates a key dynamically along with integrity check parameters [15].

In symmetric key cryptography, data can be encrypted either individually (stream cipher) or in group (block cipher). In stream cipher plaintext is encrypted or decrypted one symbol or character at a time with corresponding one symbol or character of key stream. On the other hand, block cipher takes 'n' bits (group of symbols) as an input and produces 'n' bits output block [16]. Block cipher is more secure and efficient than stream cipher [17]. Modern block cipher uses invertible, self-invertible and non-invertible components to encrypt data. It includes D-box, S-box, Exclusive-Or, Circular shift, Swap, Split and combine [4]. Based on these cipher components, block cipher is categorized in two types, feistel cipher and non-feistel cipher. Feistel cipher divides the input plaintext block into two halves, each half passes through different number of rounds for encryption and then combines to produce the cipher text block [18]. Non-feistel cipher uses only invertible components. There is no need to divide the block into two halves. Advance encryption standard (AES) is an example of non-feistel cipher. AES uses 128-bit block to cipher the plain text using 128, 192 and 256 bits of key. AES has been introduced into three versions. These versions can be distinguished on the bases of key size and number of rounds. AES-128 uses 10 rounds and 128-bit key, AES-192 uses 12 rounds 192 bit key and AES-256 uses 14 rounds and 256-bit key [19]. Each round of AES consists of four transformations i.e. byte substitution, shift row, mixed column and add round key [20].

Modes of operation are used to encipher plaintext larger than the block size. The Message of arbitrary length can be ciphered using a mode of operation for block cipher. The

Raza Ali and Muhammad Ali are with Electrical Engineering Department, University of Engineering & Technology (UET), Lahore, Pakistan.
Email: razaali.te@gmail.com, m.ali@uet.edu.pk. Manuscript received on Feb 04, 2017 revised on June 28, 2017 and accepted on Nov 14, 2017

most common modes of operations for block cipher include Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB) and Counter (CTR) [21]. The Counter mode of operation is slightly different from other modes as it uses counter in encryption/decryption. Set of input blocks (counter blocks) are encrypted using the key which results in a sequence of output blocks (key stream blocks). These output blocks are XORed with the plaintext and produces the cipher text. Decryption is opposite to encryption as ciphertext is XORed with the key stream to produce the plaintext [22]. CTR creates n-bit cipher text blocks, independent from each other but dependent on counter value. It cannot be used for real time processing and need complete n bit block before encryption [21].

II. ALGORITHM

The designed algorithm lays in the category of symmetric key algorithm. This algorithm is based on non-feistel block cipher which takes 128 bits plain text as an input and generates the cipher text of the same number of bits.

A. *Encryption*

The encryption algorithm, Fig. 1, is consist of four transformations namely, matrix transposition, average value, bit level circular shifting, key and counter XORing.

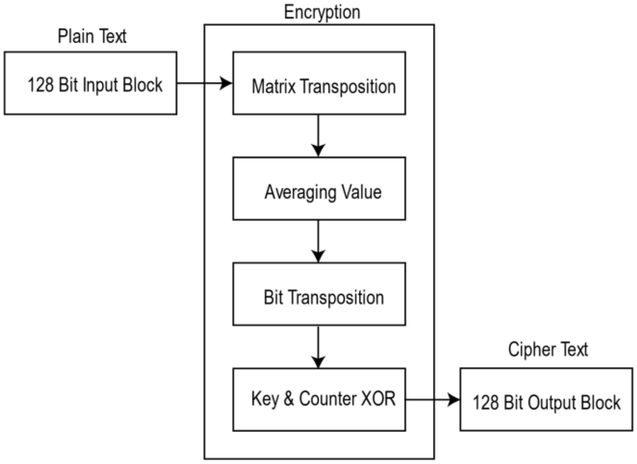


Fig.1. Encryption

1) *Matrix Transposition*

Matrix transposition takes 128-bit and converts these 128 bits into 16 characters each of 8 bits. These 16 characters are used to form a matrix of 4 x 4. In matrix transposition, firstly the left diagonal elements interchange their positions as the element ‘A’ interchanges with ‘F’ and ‘B’ interchanges with ‘E’. In second part route cipher of ‘I’ pattern is used to juggle the elements position. ‘I’ pattern starts from element ‘C’ and finishes at ‘E’. In third part the right diagonal elements interchange their positions as element ‘H’ interchange with ‘L’ and ‘K’ interchanges with ‘G’. The matrix transposition is shown in Fig. 2.

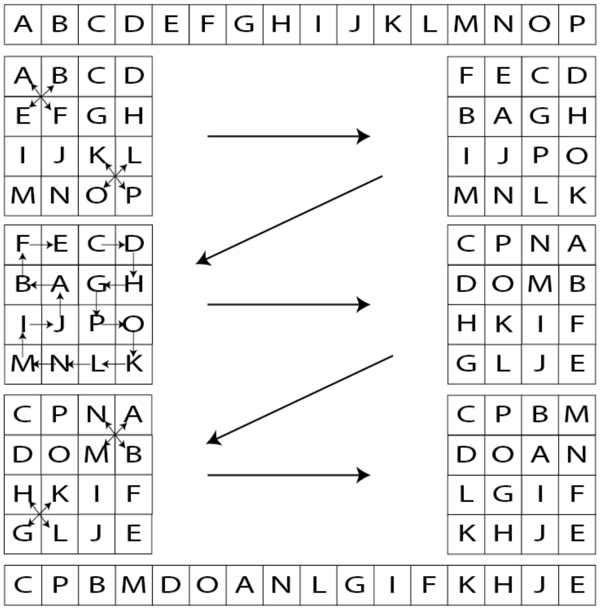

A B C D E F G H I J K L M N O P
A B C D / E F G H / I J K L / M N O P
F E C D / B A G H / I J P O / M N L K
C P N A / D O M B / H K I F / G L J E
C P B M / D O A N / L G I F / K H J E
C P B M D O A N L G I F K H J E

Fig. 2. Left diagonal Matrix transposition, I pattern Route cipher and right diagonal matrix transposition

2) *Average Value*

After right diagonal transposition plaintext is arranged in a single row matrix. Each character’s decimal value is added together and divided by 16. The average value is then added to each individual decimal value of plain text taken after right diagonal transposition. The new decimals values are then converted into their respective ASCII characters Fig. 3.

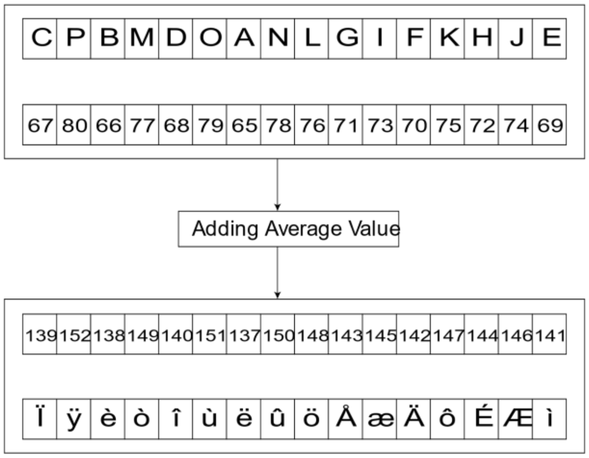


Fig. 3. Adding of average value

3) *Bit Level Transposition*

In bit level transposition, round key is used. Each ASCII characters of plaintext is picked against each key character i.e., 16-byte plain text block and 16-byte key. Charters of plaintext and key are converted into their respective 8 bits binary. The three right most bits of each individual character of the key is used to calculate decimal value and then their respective plaintext characters bits are shifted right accordingly this decimal value. Similarly, bits of next character of plain text are shifted according to second character of round keys decimal value of three right most bits.

4) *Byte level Substitution Cipher*

Mode of block cipher used in this algorithm is counter mode. In substitution cipher complement of 128-bit of round key is XOR with 128-bit counter value and the resultant 128-bit is XOR with plain text produces the final cipher text Fig. 4.

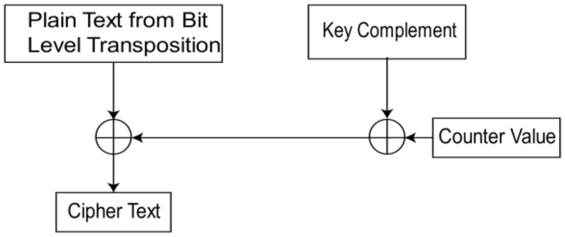


Fig. 4. Byte level transposition

B. *Decryption*

The decryption algorithm is reverse process of encryption algorithm, shown in figure 5. In decryption algorithm, complement of key is XORed with counter value and then the resultant stream of bits is XORed with the received cipher text. In next step, bits of each byte are right shifted according to the decimal value of three most significant bits of each byte of key.

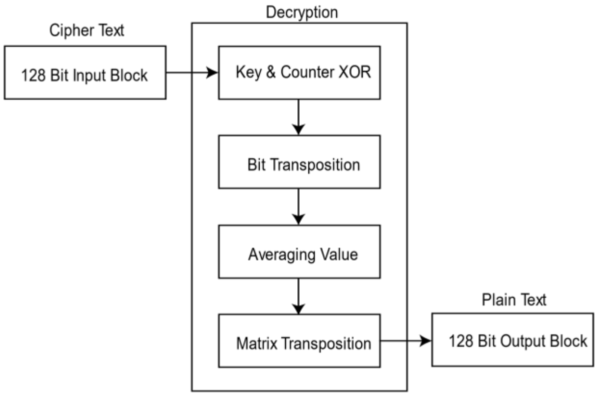


Fig. 5 Decryption

In next step, average value is calculated and divided by 2. Then the resultant value is again divided by 16 i.e. total numbers of bytes in a block. The resultant value is minus from the decimal values to get the input for last cipher stage of decryption algorithm in which block is transformed into matrix of 4 x 4. In inverse matrix transposition, first right diagonal interchanges then inverse 'I' pattern and in last left diagonal interchanges operation is performed to get the plain text.

C. *Key Generation*

In this algorithm, a unique and complex method is used to generate round key from the user defined key, Fig. 6. The original user defined 16-byte key is processed in such a complex way that the generated round key is completely different from the original key. Each byte of the original key is XORed with the next adjacent byte and last byte is XORed with first byte. These modified 128 bits are divided into 4 parts each of 32 bits. The first four bytes make block A, next four bytes makes block B, next four bytes makes block C and last byte makes block D.

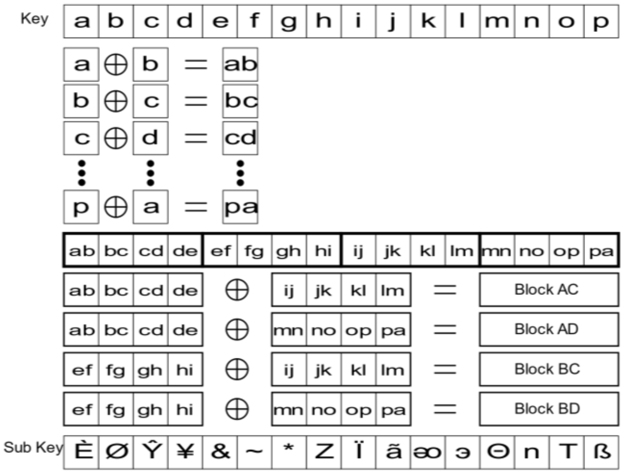

Key
a b c d e f g h i j k l m n o p
a ⊕ b = ab
b ⊕ c = bc
c ⊕ d = cd
p ⊕ a = pa
ab bc cd de ef fg gh hi ij jk kl lm mn no op pa
ab bc cd de ⊕ ij jk kl lm = Block AC
ab bc cd de ⊕ mn no op pa = Block AD
ef fg gh hi ⊕ ij jk kl lm = Block BC
ef fg gh hi ⊕ mn no op pa = Block BD
Sub Key
È Ø Ŷ ¥ & ~ * Z Ï ã ɶ э Θ n T ß

Fig. 6 Key generation method

Block A is XORed with block C makes new block AC. Block A again XORed with block D makes new block AD. Now block B XORed with block C and D to produce new block BC and BD. All new blocks AC, AD BC and BD are then combining to produce round key.

D. *Counter Value*

Each block includes counter value to accomplish the encryption process. Mode of block cipher used in this algorithm is counter mode as each time new block encrypts with new counter value. The process of generating counter value is again unique. Two initial counters of 6 byte each are used, the user and these two initial counters generate each initial vector generates 128-bit counter value concatenation with four bytes produces because of sum of decimal values of the key, as shown in figure 7.

Initial Counter 1 (6 Bytes)	Decimal Sum of Key (4 Bytes)	Initial Counter 2 (6 Bytes)

Fig. 7 Counter Value

III. ANALYSIS & TEST RESULTS

The proposed algorithm is analyzed using different cryptanalysis techniques such as linear cryptanalysis, differential cryptanalysis, brute force attack, frequency attack and avalanche effect. For analysis and results personal computer is used with following specifications:

Table 1. Personal Computer Specifications

Processor	Intel(R) Core i3-3220 CPU 3.3GHz
RAM	4.00 GB (3.41 Usable)
Hard disk	500 GB
OS	Window 7 32-bit

A. *Brute Force Attack*

Brute force attack is type of attack, in which the attacker or intruder tries every single possible key or plaintext to decrypt the cipher text or uses all combinations to sort out the plaintext or key. Some algorithms are immune from this attack such as AES [23]. Exhaustive key search needs 2^n operations to find the appropriate key where n is the number of bits. The number of key bits used in this algorithm is 128, so the number of combinations is $2^{128} = 3.04 \times 10^{38}$. The fastest computer can perform 10.51×10^{15} FLOPS (Floating point Operations per Second) [24].

$$No. of\ FLOPS\ per\ combination = 1000\ (be\ optimist)$$

$$No. of\ combination\ check\ per\ second = \frac{10.51 \text{ x } 10^{15}}{1000} = 10.51 \text{ x } 10^{12}$$

$$Second\ per\ year = 365\ \text{ x } 24 \text{ x } 60 \text{ x } 60 = 31536000 \text{ sec}$$

$$\text{Combinations of } 128 \text{ bits} = 2^{128} = 3.4028 \text{ x } 10^{38}$$

$$No. of\ years\ to\ break\ 128\ bit\ key = \frac{3.4028 \text{ x } 10^{38}}{(10.51 \text{ x } 10^{12})(31536000)}$$
$$= 1.02 \text{ x } 10^{18}\ years\ [xvii]$$

If the no. of FLOPS becomes 33.86 PFLOPS $(33.86 \text{ x } 10^{12})$

$$No. of\ years\ to\ break\ 128\ bit\ key = \frac{3.4028 \text{ x } 10^{38}}{(33.86 \text{ x } 10^{12})(31536000)}$$
$$= 3.18 \text{ x } 10^{17}\ years$$

The Indian Institute of Science has planned to build the fastest computer in 2017 and they proposed to achieve 132.6 EFLOPS (132.8 X10^{18}). For this type of computer,

$$No. of\ years\ to\ break\ 128\ bit\ key = \frac{3.4028 \text{ x } 10^{38}}{(132.8 \text{ x } 10^{15})(31536000)}$$
$$= 8.11 \text{ x } 10^{13}\ years$$

In this algorithm the key length is 128-bit so to apply brute force attack, the intruder needs to process for minimum 8.11×10^{13} years with 132.8 x 1018 FLOPS. The length key bits are large enough to avoid exhaustive key search. Intruder must try all the possible combinations which require nearly 2^{128} operations to determine the key, which is practically infeasible. So, this algorithm is immune to brute force attack.

B. *Frequency Attack*

Frequency analysis or frequency attack is based on the probability of occurrence of letter which may be single letter, double letters, or triple letters in any language.

In cryptography frequency analysis is a very powerful tool used for cryptanalysis. Using this tool an intruder can find the encrypted information hidden in cipher text. Each letter in English language has probability of occurrence, called frequency of occurrence of that letter. This frequency can be used on cipher text to guess the plaintext.

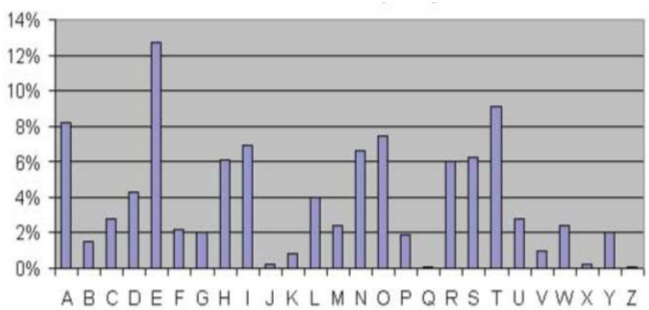
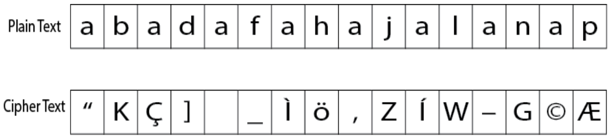


Fig. 8 Relative Frequency Graph [25]

The main idea behind the use of this frequency attack is to calculate the probability or frequency of occurrence of letters present in cipher text and compare them with guessed plain text letters. Figure 8 shows the distribution table on frequency of letters in English language [25].

In this proposed algorithm the frequency analysis is not useful because the ciphering process consists of substitution based cipher, transposition based cipher (both bit and byte level) and average of decimal value which truly eliminate the effect of frequency analysis, as shown in Figure 9.



Plain Text	a	b	a	d	a	f	a	h	a	j	a	l	a	n	a	p
Cipher Text	“	K	Ç	]		_	Ì	ö	,	Z	Í	W	–	G	©	Æ

Fig. 9 Frequency analysis using 8 times ‘a’ in a single block

The cipher text is also in the range of extended ASCII 0-255. The extended ASCII has almost 127 characters or symbols other than capital letters, small letters, numbers, space, and other punctuation marks. So, it also makes frequency attack useless for this algorithm. Figure 10 and 11 show the frequency of occurrence of each character of plaint text and cipher text. The letter “a” is used 8 times in plain text but none of the character or symbol in cipher text is repetitive.

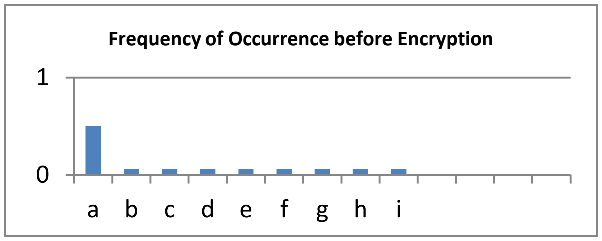


Fig. 10 Frequency of occurrence of Plain text characters

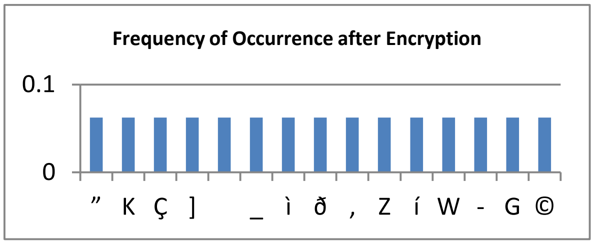


Fig. 11 Frequency of occurrence of Cipher text characters

C. Avalanche Effect

In cryptography, to design or analyze any algorithm, avalanche effect must be considered. It is one of the desired properties for any successful algorithm.

Avalanche effect is evident, when slight change in an input of encryption algorithm gives a significant amount of change in output.

The avalanche effect is calculated using the following formula:

$$Avalanche\ Effect(\%) = \frac{No.of\ changed\ bits\ in\ cipher\ text}{Total\ bits\ in\ cipher\ text} \times 100$$

Change in one bit either in plaintext produce a drastic amount of difference in cipher text.

Table 2. Avalanche effect (Bytes)

Key	IV I	IV II	Plaintext	Cipher text
abcdefghijklmnop	123456	987654	1234567891011122	IFGDIBivGzQ|[zuD
abcdefghijklmnop	123456	987654	123456789101112T	qõOLqJa~ObYtCb}L
abcdefghijklmnop	123456	987654	123456789101112a	u sH}vezKf]hOnyH
abcdefghijklmnop	123456	987654	123456789101112z	ywtyrYFwj!lKjET
abcdefghijklmnop	123456	987654	12@4567891011122	JK@uNmrC~UpGfq@
abcdefghijklmnop	123456	987654	12&4567891011122	±BCXMFuj[vMx_~ix

The avalanche effect is considered one of the important parameters for any algorithm to be secure and the proposed algorithm shows good results. As change in bit of plaintext produces at least 35 changed bits and 100% different cipher text of 16 bytes.

D. Throughput Time

The encryption time of different text block is calculated to find the throughput of the encryption algorithm.

Table 3. Avalanche Effect (Bits)

1st Input Text	**long live Pak092**
1st Input Bits	01101100011011110110111001100111001000000110110001101001011101100110010100100000010100000110000101101011001100000011100100110010
1st Output Bits	11011111011101111010010000101011010101100000010111110110000001111000010101110000101000001001001100100110101010001101100010001011 0
1st Output Text	**ßw¤+VöáA&MQ±**
2nd Input Text	**l/ng live Pak092**
2nd Input Bits	01101100001011110110111001100111001000000110110001101001011101100110010100100000010100000110000101101011001100000011100100110010
2nd Output Bits	00101110011101011110000001010111010100100100010100110001100011110001101011100011100000010101101001001001100100011111000110010110
2nd Output Text	**.uàWRE1_ ãZIñ**

The throughput is calculated using the following formula:

$$Throughtput = \frac{Total\ plaintext\ in\ bytes}{Encryption\ time\ (sec)} = \frac{16}{0.005662} = 2826\ Bytes/Sec$$

Table 4. Encryption time

Key	IV I	IV II	Plaintext	Encryption time (Sec)
PakistanPakistan	789456	123456	abcdef1234567890	0.006341 sec
PakistanPakistan	789456	123456	!@#$%^&*(){}:?><	0.005219 sec
PakistanPakistan	789456	123456	abcdefghijkmnop	0.007462 sec
PakistanPakistan	789456	123456	ABCDRFGHIJKLMNOP	0.004450 sec
PakistanPakistan	789456	123456	0812459875632145	0.005843 sec
PakistanPakistan	789456	123456	IJNG$%^&P0846382	0.004924 sec
PakistanPakistan	789456	123456	ðlÎ@*(¶oE XPyª1Q	0.006659 sec
PakistanPakistan	789456	123456	@@@@@@@@@@@@@@@@	0.004961 sec
PakistanPakistan	789456	123456	AAAAAAAAAAAAAAAA	0.004864 sec
PakistanPakistan	789456	123456	mmmmmmmmnnnnnnnn	0.005903 sec
Average time (sec)				0.005662 sec

The through put calculated for this algorithm is 2826 bytes per second which can be further increased, because the calculation is made using personal computer, where several other processes are also executing simultaneously on backend.

IV. COMPARISON

Table 5 shows the general comparison between AES and proposed algorithm. Algorithms are compared on the bases of brute force attack, frequency attack, avalanche effect, encryption time and throughput value.

Table 5. Comparison

Attacks	AES (128 bit)	My Algorithm	Remarks
Brute Force Attack	1.02 x 10^{18} years	1.02 x 10^{18} years	Both uses128 Bits
Frequency Attack	Good	Good	Both algorithm has immunity against Frequency attack
Avalanche Effect	Good	Good	Both algorithm has high Avalanche effect
Simulation time	0.0585 sec	0.005662 sec	Proposed algorithm has small encryption time
Throughput	273 Bytes/Sec	2826 Bytes/Sec	Proposed algorithm has high throughput value

V. CONCLUSION

The work presented in this paper attempts to develop a new encryption/decryption algorithm which is based on combination of four different ciphers which includes matrix transposition, adding average value, circular shift and key-counter XOR. The operations of these ciphers are both bit level and byte level transformation which obscures any type of relation between plaintext, key and cipher text. The Block size of plaintext, selected for this algorithm is 128-bit which is secure against brute force attack. Key size is also 128-bit and can be considered secure against exhaustive key search attack.

The proposed algorithm is analyzed against different attacks such as brute force attack, frequency attack, avalanche effect and throughput time and has shown strong immunity against them. Each cipher has individual impact on encryption and as a combination provides good encryption.

REFERENCES

[1] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2 ed.: John Wiley and Sons, November 1995.

[2] A. Nadeem and M. Y. Javed, "A Performance Comparison of Data Encryption Algorithms," in Information and Communication Technologies, 2005. ICICT 2005. First International Conference on, 2005, pp. 84-89.

[3] V. M. V. S.Sudha, K.Brindha, L. Agilandeeswari, G.Ramya, "Implementation of Enhanced Data Encryption Standard on MANET with less energy consumption through limited computation," In Proceeding of International Journal of Engineering Research and Development, vol. 2, pp. 46-52, July 2012.

[4] Xin Zhou, Xiaofei Tang, Research and Implementation of RSA Algorithm for Encryption and Decryption, 2011 the 6th International Forum on Strategic Technology.

[5] YANG Xiao-guang. Analysis about ECC Algorithm and Program Realization[J]. vol. 3, September 2004, pp. 13-16

[6] B. A. Forouzan, Cryptography and Network Security: McGraw-Hill, 2008.

[7] C. Lanxiang and Z. Shuming, "The comparisons between public key and symmetric key cryptography in protecting storage systems," in Computer Application and System Modeling (ICCASM), 2010 International Conference on, 2010, pp. V4-494-V4-502.

[8] L. R. D. Thomas Hardjono, Security In Wireless LANS And MANS: Artech House, 2005.

[9] N. Khanna, J. Nath, J. James, S. Chakraborty, A. Chakrabarti and A. Nath, "New Symmetric Key Cryptographic Algorithm Using Combined Bit Manipulation and MSA Encryption Algorithm: NJJSAA Symmetric Key Algorithm," 2011 International Conference on Communication Systems and Network Technologies, Katra, Jammu, 2011, pp. 125-130.

[10] D. Trinca, "Sequential and Parallel Cascaded Convolutional Encryption with Local Propagation: Toward Future Directions in Symmetric Cryptography," Third International Conference on Information Technology: New Generations (ITNG'06), Las Vegas, NV, 2006, pp. 464-469.

[11] M. M. Abdelwahab, "Encryption implementation based on symmetric algorithm using a key rounds upto four rounds," 2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), Khartoum, 2015, pp. 27-31.

[12] M. J. M. Chowdhury and T. Pal, "A New Symmetric Key Encryption Algorithm Based on 2-d Geometry," 2009 International Conference on Electronic Computer Technology, Macau, 2009, pp. 541-544

[13] V. Mullick, "E-HATS: A symmetric key encryption algorithm with dynamic substitution rule," 2014 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, 2014, pp. 1-6.

[14] D. H. Kurniawan and R. Munir, "Double Chaining Algorithm: A secure symmetric-key encryption algorithm," 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), George Town, 2016, pp. 1-6.

[15] J. Narayanaswamy, R. V. Sampangi and S. Sampalli, "HIDE: Hybrid symmetric key algorithm for integrity check, dynamic key generation and encryption," 2015 International Conference on Information Systems Security and Privacy (ICISSP), Angers, 2015, pp. 124-131.

[16] U. P. a. M. M. a. J. Jain, "Article: A Novel Approach for Image Encryption by New M Box Encryption Algorithm using Block based Transformation along with Shuffle Operation," International Journal of Computer Applications, vol. 42, pp. 9-15, 2012.

[17] G. S. Vishwa gupta, Ravindra Gupta, "Advance cryptography algorithm for improving data security," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, 2010.

[18] W. Stallings, Cryptography and Network Security: Principles and Practice, 5 ed.: Pearson, 2010.

[19] "Advanced Encryption Standard," [online] 2014, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard (Accessed: 8 August 2014).

[20] "NIST, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 26, 2001, available at: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf."

[21] S. M. Diesburg, C. R. Meyers, D. M. Lary, and A.-I. A. Wang, "When cryptography meets storage," presented at the Proceedings of the 4th ACM international workshop on Storage security and survivability, Alexandria, Virginia, USA, 2008.

[22] R. Tirtea and G. Deconinck, "Specifications overview for counter mode of operation. Security aspects in case of faults," in Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean, 2004, pp. 769-773 Vol.2.

[23] "Security of AES against brute force attack," [online] 2014, http://www.eetimes.com/document.asp?doc_id=1279619 (Accessed: 12 October 2014).

[24] "Floating-Point Operation per Second," [online] 2014, http://en.wikipedia.org/wiki/FLOPS (Accessed: 3 October 2014).

[25] "Frequency Analysis," [online] 2014, http://en.wikibooks.org/wiki/Cryptography/Frequency_analysis (Accessed: 8 October 2014).