Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

11th International Conference
on Electrical Engineering
ICEENG 2018

Design and Assessment of a Novel Encryption Algorithm for a Special Purpose Virtual Private Networks

Hany Ahmed Ramzy [2] , Mohamed Hussein [1], Alaa E. Omar [2], Khaled Shehata

The Virtual Private Network (VPN) technology provides a way of protecting information being transmitted over unsecure network. VPN uses encryption to provide data confidentiality and data integrity but does not provide or enforce strong user authentication. This paper proposes a new symmetric encryption algorithm for special purpose VPN to secure the classified data. The proposed algorithm based non Feistel structure with well-designed substitution boxes (S-Box), permutation networks, and a serial-parallel construction functions, in addition to key derivation algorithm which used the input secret key to generate sub-keys input for proposed algorithm. The security analysis of the proposed was conducted and the National Institute of Standards and technology statistical tests have been applied to the output of the algorithm proving its secrecy and randomness properties. In addition, the performance of the proposed algorithm has measured per the execution time comparing with the standard algorithm.

KEYWORDS

Virtual Private Network, Substitution Boxes, Secrecy.

* Egyptian Armed Forces.

1. INTRODUCTION

VPN (Virtual Private Network) is a generic term used to describe a communication network that uses any combination of technologies to secure a connection tunneled through an otherwise unsecured or un-trusted network. Instead of using a dedicated connection, such as leased line, a "virtual" connection is made between geographically dispersed users and networks over a shared or public network [1-3]. Once connected, the VPN makes use of the tunneling mechanism to encapsulate encrypted data into a secure tunnel, with openly read headers that can cross a public network. Internet protocol security (IPsec) was developed by the Internet Engineering Task Force (IETF) for secure transfer of information at the OSI layer three across a public unprotected IP network, such as the Internet. IPsec makes use of two security protocols, Authentication header (AH) and Encapsulated Security Payload (ESP), for required services. However, IPsec is limited to only sending IP packets. ESP uses standard symmetric encryption algorithms, such as AES, to provide data privacy which becomes (eventually) insecure to use with the classified data in a special purpose VPN [4-8].

For a stream cipher implementation to remain secure its pseudorandom generator should be unpredictable and the key should never be reused. Stream ciphers are designed to approximate an idealized cipher, known as the One-Time Pad. The One-Time Pad, which is supposed to employ a purely random key, can potentially achieve "perfect secrecy". That is, it's supposed to be fully immune to brute force attacks [9-11].

The proposed algorithm based non-Festal structure with well-designed substitution boxes (S-Box), permutation networks, and a serial-parallel construction functions, in addition to key derivation algorithm which used the input secret key to generate sub-keys for the proposed algorithm. The security analysis of the proposed was conducted and the NIST statistical tests have been applied to the output of the algorithm proving its secrecy and randomness properties. In addition, the performance of the proposed algorithm has measured per the execution time comparing with the standard algorithm.

This paper is structured as follows; Section 2 introduces the proposed encryption algorithm. Then Section 3 shows the security analysis applied to the proposed algorithm. Section 4 shows the design and implementation. Finally, Section 5 concludes the paper and indicates future work.

2. The Proposed Encryption Algorithm

The proposed algorithm is divided into two main parts; the first part is related to the key derivation algorithm and the second part related to the encryption process. Our proposed algorithm aims to achieve maximum linear complexity of the output sequence and prevent leakage to avoid any modularizing attack directed towards the sub modules of the driving subsystem of the key stream generator requirements by introducing more than one high nonlinear part such as multiplexers, nonlinear reduction selection functions and combination functions.

2.1 The Key Derivation Algorithm

The goal is to produce a list of keys for additional rounds stage of the encryption algorithm, This stage consists of four registers of 32 bit connected in series, function F consists of four different S-Boxes based on inverse function in GF(28) (to increase the non linearity), followed by a linear transformation matrix in order to apply diffusion to the output keys, One XOR operation, as shown in Fig.1. The input to the stage is 128 bit in the form of secret key; the output is a list of 11 keys each of length 128 bits applied to encryption algorithm. The main tasks of our design are Shift operation design and the F function design.

2.1.1 F Function Design

The input of the function is 32 bit word coming from register 1,The output is 32 bit word applied to the XOR as shown in Fig. 2, This function consists of :-

- S-Boxes to increase the nonlinearity of the output.
- Linear Transformation matrix (32x32) to apply diffusion to the output.

The 32 bit word input is divided into 4 bytes each byte applied to a S-Box. The 32 bit output from the S-Boxes is applied to a linear transformation matrix (32x32) to perform diffusion to produce 32 bit output which will be applied to the XOR.

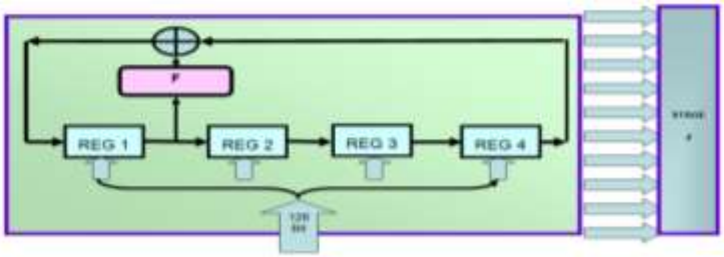

Fig.1. the Proposed Algorithm

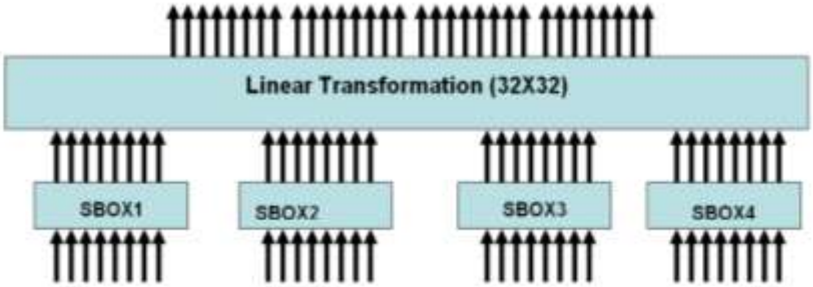

Fig.2. the F Function Design

2.1.2 Linear Transformation operation

The Linear transformation operation as shown in Fig.3, consists of (1-D) matrix of length 32 for the s-box output (A) and (2-D) matrix (B). Each bit from matrix A will be AND with each bit from the first row of the matrix B. The outputs will be XORed to produce the first bit of the 32 bit output. Then the same operation will be done between each bit of the matrix A and all the remaining 31 row of matrix B to get the 32 bit output.

$$\overset{B}{\begin{pmatrix} b_{(1,1)} \ldots\ldots b_{(1,32)} \\ \\ b_{(32,1)} \ldots\ldots b_{(32,32)} \end{pmatrix}} \overset{A}{\begin{pmatrix} a_{1} \\ \\ a_{32} \end{pmatrix}} = \overset{op}{\begin{pmatrix} op_{1} \\ \\ op_{32} \end{pmatrix}}$$

$$Op_{(i)} = [\ (a_{(i)}\ \&\ b_{(i,j)}) \oplus (a_{(i+1)}\ \&\ b_{(I,j+1)}) \oplus \ldots\ldots \oplus (a_{(i+32)}\ \&\ b_{(I,j+32)})\ ]$$

Fig.3. the Linear Transformation Function

The Key Derivation algorithm flow chart is shown in Fig. 4. At the start, the SPC register contents are initialized to zeros. Upon receiving the 4 words from the key we set the Number of shifts to 0, and the number of derived key to 0. We assign the register contents with the input key, then make shift. After 4 shifts we can Issue the key, then increments the number of issued keys until we issue 10 keys.

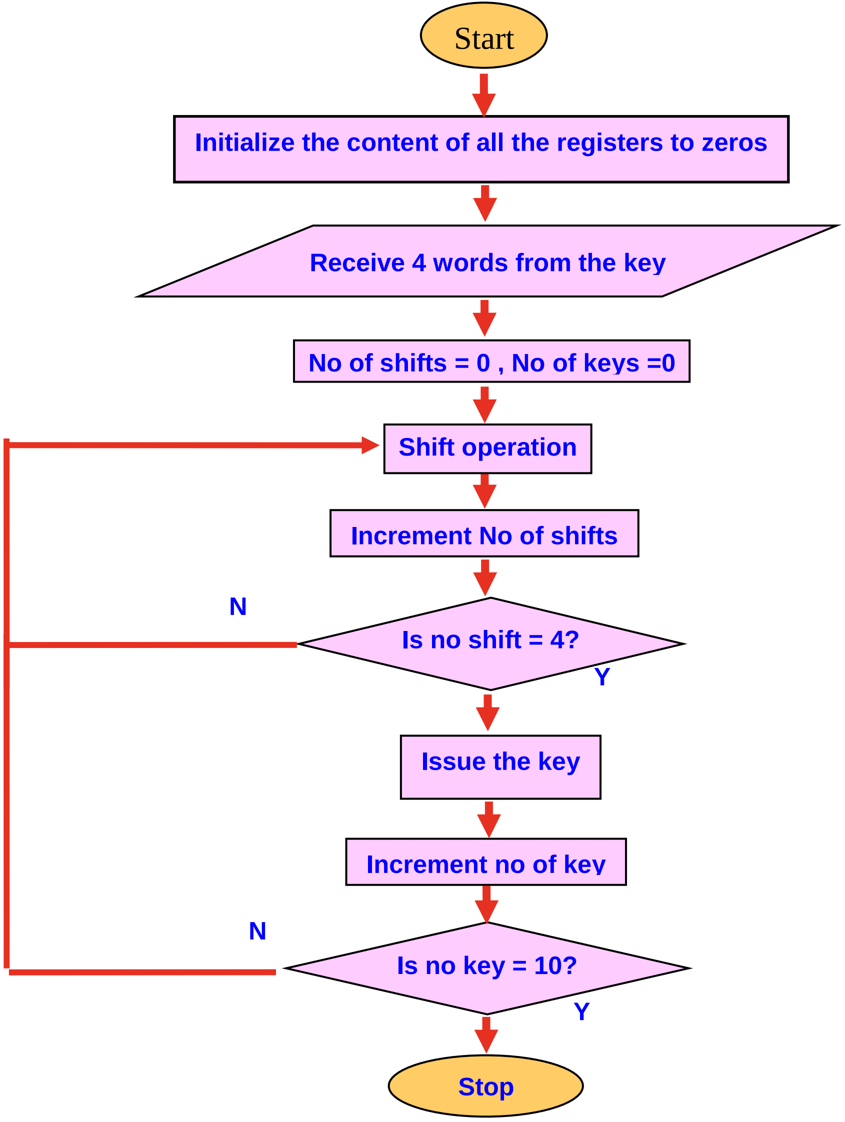

Fig. 4. the Key Derivation Algorithm Flow Chart

2.2 The Proposed Encryption Algorithm

The proposed encryption algorithm consists of a nonlinear input register, **S**erial and **P**arallel **C**onstruction **SPC**, consists of 37 shift registers, each register consists of 7 cells (259 cells).We used (8) S-boxes in the design to increase the complexity and the non-linearity. Also the multiplexer function is used at the end of the proposed algorithm to choose which output will take from the 4 S-boxes. The output of the multiplexer is the key stream that will be XORed with the plain text to produce the cipher text, as shown in Fig. 5.

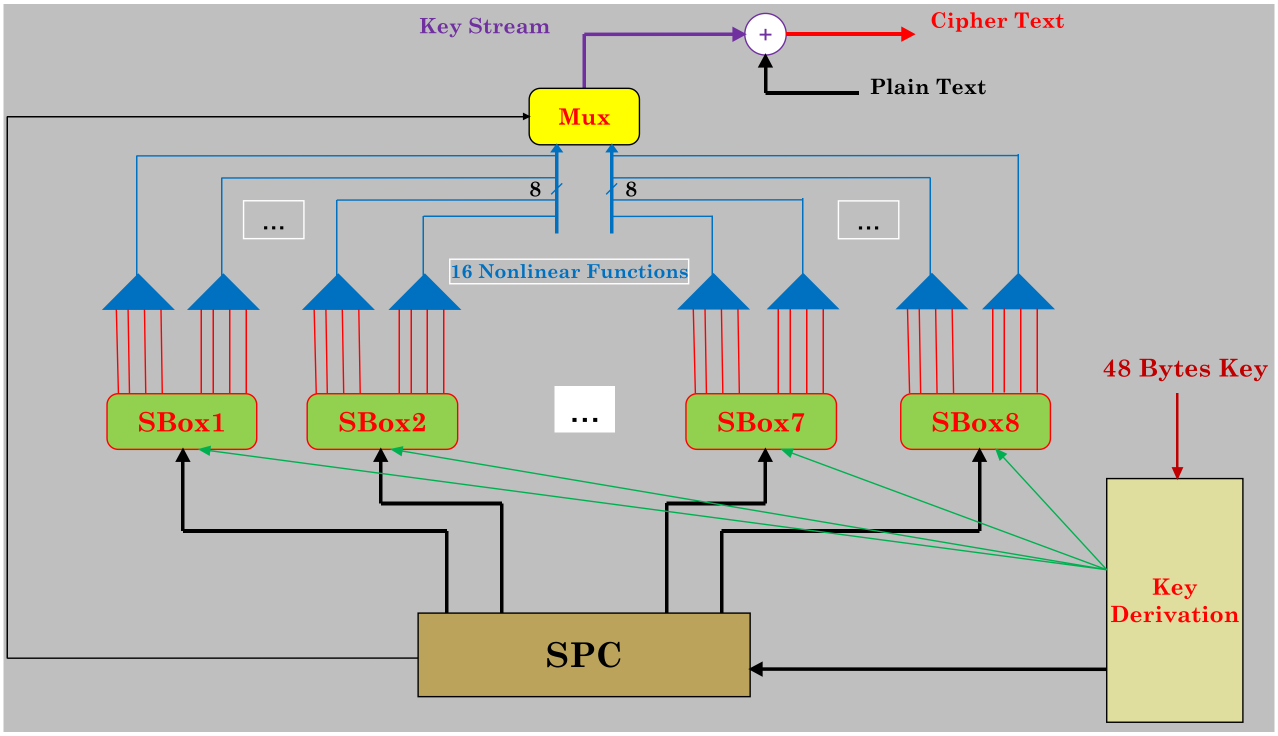

Fig.5. the Proposed Encryption Algorithm Block Diagram

The basic building block construction: Shift register, Two XOR gates, Boolean function (T_Boolean), and Expansion function (F_Key).

We have 6 input Boolean to this function the number of possible Boolean function = 10^19 Boolean function we will draw simple structure example (4 and gate + 1 or gate), each gate has 3 input. Inside this family of function we find tow ways to obtain a balance use for the 1st 2 bits of the input bits as selector to all the rest of the input bits (multiplexer) .

Applied the output to the next state following the next construction (8 S boxes and multiplexer), the 8 S-boxes and 16 nonlinear function will used to increase the non-linearity concept and support the confusion and diffusion concept to make the relation between the plain text, key and cipher text as far as possible, where the 8 bit output will apply to multiplexer with the selector from the output of SPC, hence the plain text will XORed with output stream of bits to get cipher text.

Applied the output of the serial parallel construction to the next state following the next construction (8 S boxes and multiplexer), the 8 S-boxes each 256 bits and 16 nonlinear function will used to increase the non-linearity concept and support the confusion and diffusion concept to make the relation between the plain text, key and cipher text as far as possible, where the 8 bit output will apply to multiplexer with the selector from the output of SPC, hence the plain text will XORed with output stream of bits to get cipher text.

3. The Security Analysis

The proposed algorithm is designed to provide a high level of complexity, nonlinearity, randomness, balance and diffusion. Those requirements are satisfied. For example, diffusion level is enhanced by using with well-designed substitution boxes (S-Box), and a serial-parallel construction functions, in addition to key derivation algorithm.

We analyze the security of a ***Proposed Algorithm*** against various attacks. As a result, we claim that ***Proposed Algorithm*** is secure enough for cryptographic applications. The designers have not deliberately inserted any hidden weaknesses in the algorithm. In this section we will present a security analysis of ***proposed algorithm***, showing its resistance against various cryptanalytic attacks.

3.1 Brute force attacks

Secrecy of the ***Novel Encryption Algorithm*** is depending on group of unequal initial points (Substitution Box, Basic Key). According to the data types used, there are 2^{640} possible combinations for its initial points. This is equal to key length 640 bits. If we assume that there are computers works with computation power from order of 10^{20} operation per second, it will need to break $\approx 1.3 \times 10^{168}$ year to predict the secret key. Thus, Exhaustive attack [6] seems to be impractical.

3.2 Saturation Attack

The saturation attack [7] uses a saturated multi-set of plaintexts. The attacker needs the property that XOR sum of particular parts of the corresponding cipher text is zero, this is called saturation characteristic. Saturation characteristics useful for the attack are often found in block ciphers in which small portions of the bits are interleaved by a strong nonlinear function while the main interleaving stage is linear. The main interleaving stage in ***Proposed Algorithm*** is depending on non-linear functions, which avoid saturation attack.

3.3 Algebraic Attack

Algebraic attack problem due to low nonlinear degree of the combination function, as the values vary from 0000 to 1111. In order to apply the algebraic attack [8] to block ciphers, we should derive an over-defined system of algebraic equations. Since a ***Proposed Algorithm*** contains 8 S-boxes and 16 nonlinear function for increasing the non-linearity concept and support the confusion and diffusion concept to make the relation between the plain text, key and cipher text as far as possible, it may be impossible to convert any equation system in ***Proposed Algorithm*** into an over-defined system.

3.4 Statistical Random Tests

The key streams have been analyzed using the CRYPTX'98 statistical analysis program [12-13]. Seven tests are conducted; frequency test, binary derivative test, change point test, sub-block test, sequence complexity test. Test results are shown in Tables 1 to 7 and briefly described. CRYPTX'98 reported that the proposed algorithm key streams tested have the properties of random bit streams. Other CRYPTX'98 tests look for a class of statistical properties than can be exploited by various classes of attacks. The analysis reported that proposed algorithm key streams had none of these properties. This encourages us to believe that proposed algorithm has no glaring weaknesses.

The popular symmetric block stream cipher algorithms including A5 (Actual Encryption Algorithm) [14], RC4 (Rivest Cipher number four) , Rabbit, and Sober, were adapted for testing with the same CRYPTX'98 tool and their results were compared with the stream cipher output of proposed algorithm, using the same input plaintext and key files.

We apply a different cipher samples generated by implementation of proposed algorithm and other four block cipher algorithms, using C++ software programming language, to a CRYPTX'98 tool which includes randomness statistical tests and it produces a Significance probability *p*-value for each input type. The *p*-values obtained from a CRYPTX'98 test represents the probability of obtaining a result further than the test statistic lies from the expected, if the algorithm produces a random stream. Very small p-values would support non-randomness for the given measure.

3.4.1 Frequency Test

Test for an equal number of ones and zeros in the bit stream. The number of ones in a large random sequence is approximately normally distributed. The frequency test determines the tail end probability for the number of ones in the sample stream. The result is displayed on the Table 1. It can be derived from the results that none of the sequences fail the frequency test because all of the significance probabilities are all greater than 0.001. It signifies that there are an equal proportion of ones and zeros in each stream. Therefore, the sequences are considered balance and randomness. The steam cipher RC4 algorithm and then Proposed Algorithm, and A5, algorithms results seems to have better distributions, for their larger significant, than the others. Binary derivative tests should be further conducted to assess whether are patterns existing in these streams.

Table 1: Combined table comparing the results of frequency test.

	Test Parameters				
Algorithm	Number of ones (X)	Expected ones (mean)	Proportion of ones	Significance probability (*p*)	Satisfy
Proposed Algorithm	199845	20000	0.4996	0.6240	Yes
A5	199754	20000	0.4994	0.4366	Yes
RC4	200067	20000	0.5002	0.8322	Yes
Rabbit	200393	20000	0.5010	0.2140	Yes
Sober	200438	20000	0.5011	0.1660	Yes

3.4.2 Binary derivative test

The binary derivative is a new stream formed by the exclusive-or operation on successive bits in the stream. Successive binary derivative streams may be obtained from each new binary derivative, each one being of length one less than its predecessor.

The frequency test applied to the original stream and its first binary derivative is equivalent to testing for an equal number of the four overlapping two-tulles in the original stream. The frequency test applied to the original stream, its first and second binary derivatives, along with the change point test is equivalent to testing for an equal number of the eight overlapping three-tulles in the original stream.

Table 2: results of Binary Derivative test

	First binary derivative (D1) test (N = 399999)				
Algorithm Type	Number of ones (X)	Expected ones (mean)	Proportion of ones	Significance probability (p)	Satisfy
LEA	200189	199999.5	0.5005	0.5490	Yes
A5	200301	199999.5	0.5008	0.3404	Yes
RC4	200099	199999.5	0.5002	0.7530	Yes
Rabbit	200744	199999.5	0.4995	0.5158	Yes
Sober	199794	199999.5	0.5002	0.8409	Yes
	Second binary derivative (D2) test (N = 399998)				
LEA	200188	199999	0.5005	0.5501	Yes
A5	200155	199999	0.5004	0.6218	Yes
RC4	199851	199999	0.4996	0.6398	Yes
Rabbit	199717	199999	0.4993	0.3725	Yes
Sober	199705	199999	0.4993	0.3525	Yes

The result is displayed on the Table 2. It can be derived from the results that none of the sequences fail the first and second binary derivative test because all of the significance probabilities are all greater than 0.001. The equal number of ones and zeros in the first binary derivative stream indicates that the original stream contains an equal proportion of overlapping four 2-tuples, (00), (01), (10), (11). It can be observed that the stream ciphers RC4 and Sober then PROPOSED ALGORITHM and Rabbit results seems to be less patterned than the others.

3.2.3 Change point test

At each bit position in the stream the proportion of ones to that point is compared to the proportion of ones in the remaining stream. The bit where the maximum change occurs is called the 'change point'. This test determines whether this 'change' is significant. This checks that there is an equal number of overlapping three-tulles for streams which have passed the frequency test on the original stream and also on the first two binary derivatives.

Change point test result is displayed on the Table 3. There is no change in the proportion of ones throughout the whole stream. These sequences give satisfactory results to the frequency test, First Binary derivative test, second Binary derivative test, and also the change point test. They are considered to generate equal number of the overlapping 3-tuples.

Table 3: results of change point test

	Test Parameters					
Algorithm	Change point	Number of ones before	Proportion of ones before	Proportion of ones after	Significance probability (p)	Satisfy
PROPOSED ALGORITHM	222886	111150	0.4987	0.5008	0.4257	Yes
A5	265864	132539	0.4985	0.5011	0.3488	Yes
RC4	129267	64849	0.5017	0.4995	0.4716	Yes
Rabbit	215662	107625	0.4990	0.5032	0.0304	Yes
Sober	229480	115303	0.5025	0.4993	0.1432	Yes

3.2.4 Sub-block test

For sub-block sizes up to 16 the 'uniformity test' requires a sample of at Proposed Algorithms 5 * b * 2^ (b) bits, where b is the sub-block size. This test requires a sample of b * 2 ^ (b/2+3) bits. Consult the manual for recommended upper limits on the sub-block size. Sub-block test result is displayed on the Table 4. It can be observed from the results that all of the sequences satisfy the sub-block test.

Table 4: results of Sub-block test

	Test Parameters				
Algorithm	Sub-block Size	Chi-Square Value	Degree of Freedom	Significance probability (*p*)	Satisfy
PROPOSED ALGORITHM	2	0.5600	3	0.9055	Yes
A5	2	2.1209	3	0.5477	Yes
RC4	2	1.4407	3	0.6960	Yes
Rabbit	2	9.1120	3	0.0278	Yes
Sober	2	3.1295	3	0.3721	Yes

3.2.5 Running test

Test the distribution of run lengths, assuming the data satisfies the frequency test. A 'block' is a run of ones and a 'gap' is a run of zeros. The runs test compares the distribution of block and gap lengths with that expected for a binomial population in which the probability of a one is a half. This supports Golomb's postulates for which half the runs have length 1, a quarter has length 2, and an eighth has length 3, and so on. Runs test result is displayed on the Table 5. It can be observed from the results given in Table 5; all of the sequences satisfy the Runs test. The number of blocks, consecutive ones, and the number of gaps consecutive zeros, is almost equal. It indicates that there is no uniformity or such patterns in these streams.

Table 5: Results of Runs test

	Test Parameters						
Algorithm	Number of runs	Number of blocks	Number of gaps	Chi-Square Value	Degree of Freedom	Significance probability (*p*)	Satisfy
PROPOSED ALGORITHM	200190	100095	100095	21.0619	30	0.8859	Yes
A5	200302	100151	100151	24.0537	30	0.7696	Yes
RC4	200100	100050	100050	23.8395	30	0.7792	Yes
Rabbit	200745	100373	100372	33.3319	30	0.3083	Yes
Sober	199795	99898	99897	21.2072	30	0.8813	Yes

3.2.6 Sequence complexity test

Test that there is a sufficient number of new patterns encountered throughout the stream. A stream with a sequence complexity measure below a given 'threshold' value is considered non-random. An average value of sequence complexity for a stream of this length is also calculated.

Sequence complexity test result is displayed on the Table 6. It can be observed from the results given in Table 6, all of the sequences satisfy the sequence complexity test. The numbers of sequence complexity are considered as the number needed to recover the original sequence. It should be greater than threshold value and close to the mean value.

Table 6: results of Sequence complexity test

	Test Parameters			
Algorithm	Sequence complexity	Threshold Value	Mean Value	Satisfy
PROPOSED ALGORITHM	1440	1400	1415	Yes
A5	1438	1400	1415	Yes
RC4	1437	1400	1415	Yes
Rabbit	1438	1400	1415	Yes
Sober	1440	1400	1415	Yes

3.2.7 Linear complexity test

Every finite stream can be produced by a linear feedback shift register (LFSR). The linear complexity test checks for the minimum amount of knowledge (bits) needed to reconstruct the whole stream. Tests are applied to the linear complexity of the stream, and on the “"linear complexity profile” which follows the change in linear complexity as each new bit is added. Linear complexity test result is displayed on the Table 7. It can be observed from the results given in Table 7, all of the sequences satisfy the linear complexity test, all of the sequences satisfy the linear complexity-number of jumps test, and all of the sequences satisfy the linear complexity-jump size test. The expected numbers of linear complexity are considered as the half numbers of the sequence. The observed numbers of linear complexity should be as large as possible. It is derived from the Table 7, that, the stream cipher PROPOSED ALGORITHM are more complex and difficult to be detected.

Table 7: results of Linear complexity test (N=20000)

			Algorithm				
			PROPOSED ALGORITHM	**A5**	**RC4**	**Rabbit**	**Sober**
Test Parameters	Linear Complexity	LC	10002	10001	10000	10001	10000
		Expected LC	10000	10000	10000	10000	10000
		Significant Probability *P*	0.9739	0.8341	0.5000	0.8341	0.5000
	Linear Complexity Number of Jumps	Number of Jumps	5030	4988	5047	5029	4931
		Expected Number of Jumps	5000	5000	5000	5000	5000
		Significant Probability *P*	0.7257	0.4052	0.8264	0.7190	0.0838
	Linear Complexity Jump size	Chi-square value	9.1323	9.6196	8.1244	14.5572	6.2324
		Degree of freedom	8	8	8	8	8
		Significant Probability *P*	0.3313	0.2927	0.4214	0.0683	0.6212
		Satisfy	Yes	Yes	Yes	Yes	Yes

4. Design and Implementation Performance

Random behavior of ***Novel Encryption Algorithm*** is estimated in this paper by software implementation of ***Novel Encryption Algorithm***, with measuring its strength, as shown in section-3. ***Proposed Algorithm*** is not designed for notably high speeds in software, although it is straightforward to implement in reasonable efficiency. Speed comparative study between cipher algorithms was done with the known current block ciphers, implementation speed of ***Novel Encryption Algorithm*** scheme was found to be faster than AES (Advanced Encryption Standard) for all key sizes.

The popular block cipher algorithms including DES (Data Encryption Standard), 3DES (Triple DES), AES, Blowfish, were implemented, and their performance was compared with dedicated ***Novel Encryption Algorithm*** software implementation algorithm by encrypting input files of varying contents and sizes. Algorithms were implemented in a uniform programming language (C), using their standard specifications, to compare their performance.

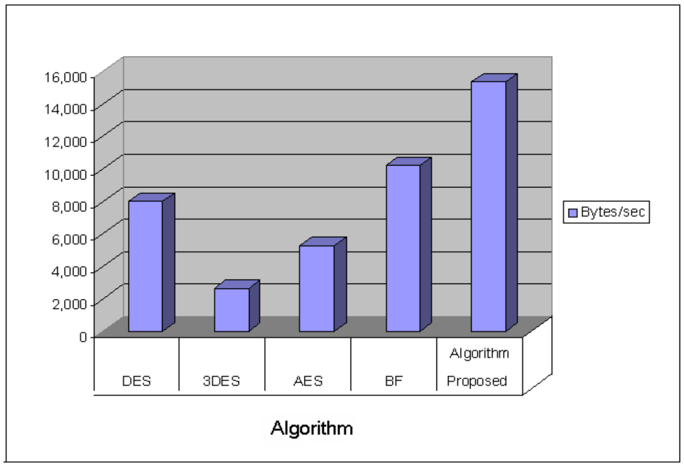

Fig. 6: Comparative Performance Evaluation (Bytes\seconds)

Table 8 shows the Comparative results of Algorithms, where they have conducted it on P-4, 2.4 GHz machine.

Table 8 : Performance Results

Input Message Size (Bytes)	**DES**	**3DES**	**AES**	**BF**	**Proposed Algorithm**
20,528	2	7	4	2	1
36,000	4	13	6	3	2
45,912	5	17	8	4	3
59,856	7	23	11	6	4
69,544	9	26	13	7	5
137,328	17	51	26	14	11
158,952	20	60	30	16	12
166,368	21	62	31	17	13
191,384	24	72	36	19	15
232,400	30	87	44	24	19
Average Time/ Samples	14	42	21	11	9
Bytes/sec	7,988	2,663	5,320	10,167	14,425

Table 8 illustrates that, ***Novel Encryption Algorithm*** cipher implementation achieves competitive performance compared with others, it can be observed that ***Novel Encryption Algorithm*** has an advantage over other algorithms in terms of throughput. Additionally, it shows also that 3DES has almost 1/3 throughput of DES, or in other words it needs 3 times than DES to process the same amount of data.

The size of the Basic key, 128, 196, or 256 bits, does not affect the speed of the algorithm because the Basic key is used only to initialize the internal key during the setup process. Also observe that the same computational effort must be invested in key setup and in key change. There are no tradeoffs between speed and memory in the implementation of ***Novel Encryption Algorithm***.

5. Conclusion

The proposed encryption algorithm satisfied the given assumptions as it passed the statistical tests and competitive performance and sometimes better performance compared with the others. It walked a step towards reducing the processing time and capacity required for applicability on resource limited environments. The complete diffusion requirement is satisfied after a single round. Further, the algorithm provides a good combination of security features. However, the extensive security analysis of any new cipher requires a lot of efforts from many researchers. We thus invite and encourage the readers to analyze the security of our algorithm. Intriguingly present has implementation requirements similar to many compact stream ciphers. We discourage the immediate deployment of present but strongly encourage its analysis.

REFERENCES

[1] J. Sen, “Applied Cryptography and Network Security”, by InTech , 2012.

[2] Asrar U. H. Sheikh, "Overview of Wireless Communications", by Springer US, Kluwer Academic Publishers, 2004.

[3] Lars R. Knudsen, "The Number of Rounds in Block Ciphers", NESSIE public reports, University of Bergen, May 12, 2000, http://www.cosic.esat.

[4] X. Zeng, C. Carlet, J. Shan, and L. Hu, "Balanced Boolean Functions with Optimum Algebraic Immunity and High nonlinearity", Cryptology ePrint Archive, Report 2010/534, 2010, https://eprint.iacr.org/2010/534.pdf.

[5] YUAN LI AND T.W.CUSICK, "STRICT AVALANCHE CRITERION OVER FINITE FIELDS", Cryptology ePrint Archive, Report 2005/361, 2005, http://eprint.iacr.org/2005/361.pdf.

[6] ZHANG Li-li, ZHANG Yu-qing, "Brute Force Attack on Block Cipher Algorithm Based on Distributed Computation", by STMOPEN.net , SCIENTIFIC TECHNICAL AND MEDICAL OPEN ACCESS JOURNALS, Computer Engineering, March 30, 2013.

[7] B. Collard, F. -X. Standaert , " A Statistical Saturation Attack against the Block Cipher PRESENT", In Topics in Cryptology – CT-RSA 2009, LNSC, Volume: 5473 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp 195-210.

[8] N. T. Courtois, "How Fast can be Algebraic Attacks on Block Ciphers”, In: Biham, E., Handscuh, H., Lucks, S.,Rijmen, V. (eds.) Symmetric Cryptography (January 07-12, 2007).

[9] A. Biryukov and D. Wagner "Slide Attacks", In: *6th International Workshop on Fast Software Encryption (FSE '99)*, Rome: Springer-Verlag, March 1999 pp.245–259, Retrieved 2007-09-03.

[10] Marina Pudovkina, "A Related-Key Attack on Block Ciphers with Weak Recurrent Key Schedules", In: Foundations and Practice of Security, Volume 6888 of Lecture Notes in Computer Science, 2012, pp 90-101.

[11] J. Kim, S. Hong, S. Lee, J. Song, H. Yang, " Truncated Differential Attacks on 8-Round CRYPTON", In: Information Security and Cryptology - ICISC 2003, Volume 2971 of Lecture Notes in Computer Science , 2004, pp 446-456.

[12] J. Lu1, W. S. Yap, and Y. Wei, "Weak Keys of the Full MISTY1 Block Cipher for Related-Key Cryptanalysis", Cryptology ePrint Archive, Report 2012/066, 2012, http://eprint.iacr.org/2012/066.pdf .

[13] E. Dawson, A. Clark, H. Gustafson and L. May, “CRYPT-X’98, User Manual”, Queensland University of Technology, 1999.

[14] Zoran Constantinescu, “Advances in Grid Computing”, by NC-SA 3.0, 2011.