***I. J. Computer Network and Information Security*, 2012, 11, 55-61**
Published Online October 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2012.11.07

A New 512 Bit Cipher for Secure Communication

[1]M. Anand Kumar and [2]Dr.S.Karthikeyan
[1]PhD Research Scholar, Karpagam University, India
[2]Asst. Professor, Department of Information Technology, College of Applied Sciences, Sultanate of Oman
anandm_ss@yahoo.co.in, skarthi@gmail.com

Abstract —The internet today is being used by millions of users for a large variety of commercial and non commercial purposes. It is controlled by different entities. It is mainly used as an efficient means for communication, entertainment and education. With the rapid growth of internet, there is a need for protecting confidential data. The Internet was however originally designed for research and educational purpose, not for commercial applications.So internet was not designed with security in mind. As the internet grows the existing security framework was not adequate for modern day applications. Cryptography play a vital role in providing security.Lot of research is going on block cipher algorithms. In this paper we present a new 512 bit block cipher named SF Block cipher. The proposed cipher is developed based on design principle known as Substitution permutation network (SP Network). The algorithm is implemented in .NET Framework and MATLAB. Cryptanalysis is carried out in the encrypted file. It was found that the encrypted file with this algorithm is difficult to break.Simulation results shows that the proposed Block cipher has better performance over other algorithms such as AES and Blowfish

***Index Terms* —** Cryptography, Encryption, Cipher, Decryption, 512 bit Block and SP Network

I. INTRODUCTION

Cryptographic algorithms plays a vital role in the field of network security There are two basic types of cryptosystems such as symmetric cryptosystems and asymmetric cryptosystems. Symmetric cryptosystems are characterized by the fact that the same key is used in encryption and decryption transformations. In contrast to symmetric cryptosystems, asymmetric cryptosystems use complementary pairs of keys for encryption and decryption transformations. One key, the private key is kept secret like the secret key in a symmetric cryptosystem. The other key, the public key, does not need to be kept secret[1]. This two key approach can simplify key management by minimizing the number of keys that need to be managed and stored in the network. The key distribution system is also much simpler as it can use unprotected medium for the distribution of the public keys. Data integrity and/or data origin authentication for a message can be provided as follows. The originator of the message generates, using all the data bits of the message contents and a secret key, an Integrity Check Value which is transmitted along with the message. The message recipient checks that the received message content and Integrity Check Values are consistent before accepting the message. A digital signature can be considered as a special case of Integrity Check Value. The digital signature may need to be used to resolve a dispute between the originator and recipient of a message. Symmetric encryption based or keyed hash algorithm based approach is usually inadequate for this purpose. Asymmetric cryptosystems provide more powerful digital signatures[2].

Many cryptographic algorithms are introduced day by day by many researchers all over the globe. But many algorithms have simple structures that can be breakable. The recent advances in cryptanalytic techniques are remarkable. A quantitative evaluation of security against powerful cryptanalytic techniques such as differential cryptanalysis and linear cryptanalysis is considered to be essential in designing any new block cipher.

In this paper a new 512 bit block cipher named SF Block cipher is proposed. The proposed cipher is developed based on design principle known as Substitution permutation network (SP Network).

II. RELATED WORK

Some of the commonly used algorithms are evaluated and implemented in C and Visual Basic to identify the weakness. Some of the block ciphers are taken into the consideration such as DES, 3DES, Blowfish, AES and RC6. Brief definitions of the most common encryption techniques are given as follows:

DES: (Data Encryption Standard), was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology).DES is (64 bits key size with 64 bits block size) . Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher [3]

3DES is an enhancement of DES; it is 64 bit block size with 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time. It is a known fact that 3DES is slower than other block cipher methods.

RC2 is a block cipher with a 64-bits block cipher with a variable key size that range from 8 to128 bits. RC2 is vulnerable to a related-key attack using 234 chosen plaintexts.

Blowfish is block cipher 64-bit block - can be used as a replacement for the DES algorithm. It takes a variable length key, ranging from 32 bits to 448 bits; default 128 bits. Blowfish is unpatented, license-free, and is available free for all uses. Blowfish has variants of 14 rounds or less. Blowfish is successor to two fish [5].

AES is a block cipher .It has variable key length of 128, 192, or 256 bits; default 256. It encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices [6]. Also, AES has been carefully tested for many security applications [2]

RC6 is block cipher derived from RC5. It was designed to meet the requirements of the Advanced Encryption Standard competition. RC6 proper has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits. Some references consider RC6 as Advanced Encryption Standard [6]

III. Problem statement

The study shows that DES, 3DES, RC2 have some security issues and performance issues. It is also found that AES and blowfish are highly secured than other encryption algorithms. The work [7] stated that "According to academic papers and reports regarding the security evaluation for such algorithms, it is difficult to ensure enough security by using the algorithms for a long time period, such as 10 or 15 years, due to advances in cryptanalysis techniques, improvement of computing power, and so on. To enhance the transition to more secure ones, National Institute of Standards and Technology (NIST) of the United States describes in various guidelines that NIST will no longer approve two-key triple DES, RSA with a 1024-bit key, and SHA-1 as the algorithms suitable for IT systems of the U.S. Federal Government after 2010". Based on this study the statement of the problem is formulated and the new algorithm is proposed.

IV. Proposed Algorithm

The proposed algorithm is formulated based on the AES algorithm SF Block cipher is a 512 bit block cipher. This Block cipher is based on a design principle known as a Substitution permutation network (SP Network). The algorithm is designed based on Advanced Standard Encryption (AES) algorithm. It takes a block of the plaintext and the key as inputs, and applies several alternating rounds or layers of substitution boxes (S-boxes) and permutation boxes (P-boxes) to produce the cipher text block. In the case of SF Block Cipher the block size is 512 bit and the key size is also 512 bit. Message block and key can be realized as a 4*16 matrix. (4 rows and 16 columns.). For encrypting and decrypting a single block, the SF Block cipher algorithm applies its Functions in N Rounds. This Round number is calculated with the following formula

Round number = (key size or block size in words) + 6

Where 1 word = 4 bytes and 6 is constant. Total round for the algorithm is based on the formula is 22. But additional 2 rounds are required and hence the total rounds for the SF Block cipher is 24 rounds. The reason is that the encryption and decryption uses the same algorithm. The length of the input messages for SF Block Cipher should be multiple of 512. If the size of the message block is smaller than 512 bit then padding is used to make the size. In this padding method, after the original last message bit, a bit "1" is inserted and then "0" bits are appended until the last message block has the length 512. If there is no space left for any extra padding "0"s, then a new 512-bit block is added at the end of the message. The initial key is derived from the user password. Round keys are generated with key expansion algorithm

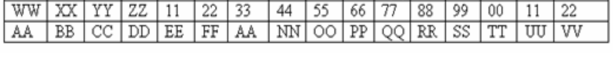
A Key Expansion algorithm

The key expansion algorithm used in the SF Block cipher is similar to that of advanced encryption standard [23]. The round keys are created from the padded key block.

B. Encryption algorithm

The encryption algorithm takes 512 bits of plaintext as the input and produces 512 bits of cipher text as an output. It consists of four steps namely sub byte round, convert row round, shifting round. and Add round key round as detailed below.

State: The input is operated in 512 bits at a time and is organized as 4 x 16 Matrix called State. Each element of the state is one byte. The plain text is arranged in a 4*16 Matrix known as state Matrix (M). The input text is converted into ASCII values as shown below.

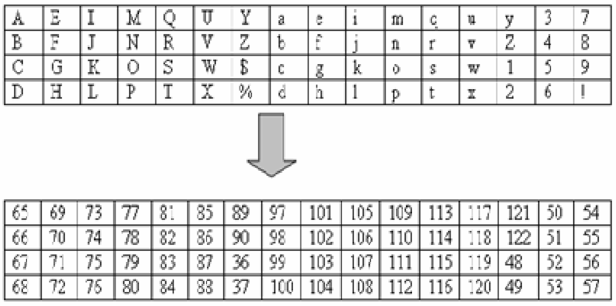


11	22	33	44	55	66	77	88	99	00	AA	BB	CC	DD	EE	FF
GG	HH	II	JJ	KK	LL	MM	NN	OO	PP	QQ	RR	SS	TT	UU	VV
WW	XX	YY	ZZ	11	22	33	44	55	66	77	88	99	00	11	22
AA	BB	CC	DD	EE	FF	AA	NN	OO	PP	QQ	RR	SS	TT	UU	VV

Figure 1: State Matrix

Sub byte round: The output of the state is given as an input to the sub byte step. In this process, the first row of key matrix is converted into binary values.16 group of 8 bit binary values are generated. The first 4-bits from each 8-bit value are separated out. Then these bits

are grouped into 16 groups such as g0……..g15. The grouped values are
{ (g0,g15), (g1,g14), (g2 ,g13), (g3,g12), (g4,g11), (g5,g10), (g6,g9), (g7,g8), (g8,g7), (g9, g6), (g10,g5), (g11,g4), (g12,g3), (g13,g2),(g14,g1),(g15,g0) }



A	E	I	M	Q	U	Y	a	e	i	m	ç	u	y	3	7
B	F	J	N	R	V	Z	b	f	j	n	r	v	Z	4	8
C	G	K	O	S	W	$	c	g	k	o	s	w	1	5	9
D	H	L	P	T	X	%	d	h	l	p	t	x	2	6	!

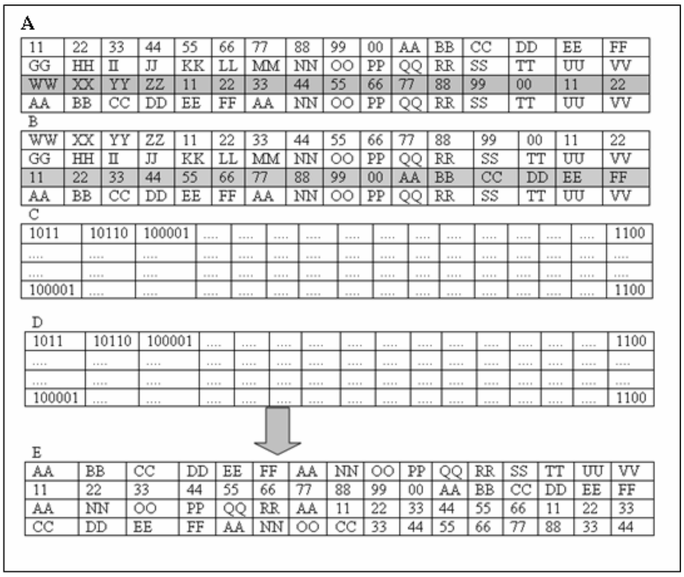
65	69	73	77	81	85	89	97	101	105	109	113	117	121	50	54
66	70	74	78	82	86	90	98	102	106	110	114	118	122	51	55
67	71	75	79	83	87	36	99	103	107	111	115	119	48	52	56
68	72	76	80	84	88	37	100	104	108	112	116	120	49	53	57

Figure 2: Sub Byte Round

The data at location M (g0, g15) is substituted from S-box. This is repeated for the remaining locations such as M(g1,g14), M(g2 ,g13), M(g3,g12), M(g4,g11), M(g5,g10), M(g6,g9), M(g7,g8), M(g8,g7), M(g9, g6), M(g10,g5), M(g11,g4), M(g12,g3), M(g13,g2), M(g14,g1), M(g15,g0) . Thus, using the first row of the cipher matrix 16 data elements are substituted in the state matrix. Similarly, the entire process is carried out using the second row of cipher key matrix and so on. After entire process the values are copied to state matrix and given to the convert row step.

Convert Row Round: In this step, each data element from the first row is represented in the hexadecimal form. After that the individual data is converted into binary form. Then read from right to left. After that data is converted back to hexadecimal form. For instance the data element is K. Its hexadecimal form is 4b. The binary form for 4b is 01001011. Then read from right to left gives 11010010. After converting it into hexadecimal, it gives D2. This process is repeated for all rows and the values are replaced.

Shifting Round: In this step the state matrix is taken from the previous step and shift operation is performed in it. Two rows are shifted. Then the output matrix is converted into binary equivalent. Now the original matrix is XoR ed with output matrix. In the following figure A is the state and B is the state after shifting row. C and D are states that are converted into Binary form. E is the final state after performing exclusive OR on C and D state.



A

11	22	33	44	55	66	77	88	99	00	AA	BB	CC	DD	EE	FF
GG	HH	II	JJ	KK	LL	MM	NN	OO	PP	QQ	RR	SS	TT	UU	VV
WW	XX	YY	ZZ	11	22	33	44	55	66	77	88	99	00	11	22
AA	BB	CC	DD	EE	FF	AA	NN	OO	PP	QQ	RR	SS	TT	UU	VV

B

WW	XX	YY	ZZ	11	22	33	44	55	66	77	88	99	00	11	22
GG	HH	II	JJ	KK	LL	MM	NN	OO	PP	QQ	RR	SS	TT	UU	VV
11	22	33	44	55	66	77	88	99	00	AA	BB	CC	DD	EE	FF
AA	BB	CC	DD	EE	FF	AA	NN	OO	PP	QQ	RR	SS	TT	UU	VV

C

1011	10110	100001	….	….	….	….	….	….	….	….	….	….	….	….	1100
….	….	….	….	….	….	….	….	….	….	….	….	….	….	….	….
….	….	….	….	….	….	….	….	….	….	….	….	….	….	….	….
100001	….	….	….	….	….	….	….	….	….	….	….	….	….	….	1100

D

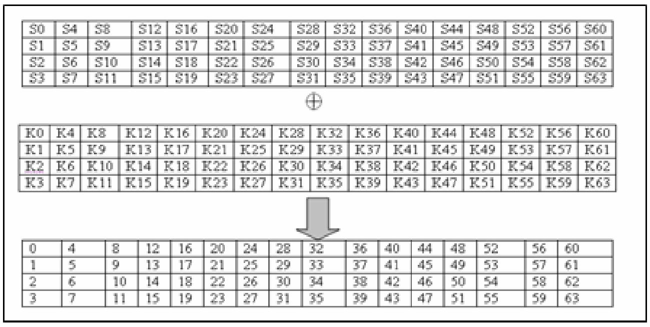
1011	10110	100001	….	….	….	….	….	….	….	….	….	….	….	….	1100
….	….	….	….	….	….	….	….	….	….	….	….	….	….	….	….
….	….	….	….	….	….	….	….	….	….	….	….	….	….	….	….
100001	….	….	….	….	….	….	….	….	….	….	….	….	….	….	1100

E

AA	BB	CC	DD	EE	FF	AA	NN	OO	PP	QQ	RR	SS	TT	UU	VV
11	22	33	44	55	66	77	88	99	00	AA	BB	CC	DD	EE	FF
AA	NN	OO	PP	QQ	RR	AA	11	22	33	44	55	66	11	22	33
CC	DD	EE	FF	AA	NN	OO	CC	33	44	55	66	77	88	33	44

Figure 3: Shifting round

Add Round Key: The actual 'encryption' is performed in the AddRoundKey () function, when each byte in the State is XORed with the sub key. The sub key is derived from the key according to a key expansion schedule



S0	S4	S8	S12	S16	S20	S24	S28	S32	S36	S40	S44	S48	S52	S56	S60
S1	S5	S9	S13	S17	S21	S25	S29	S33	S37	S41	S45	S49	S53	S57	S61
S2	S6	S10	S14	S18	S22	S26	S30	S34	S38	S42	S46	S50	S54	S58	S62
S3	S7	S11	S15	S19	S23	S27	S31	S35	S39	S43	S47	S51	S55	S59	S63

⊕

K0	K4	K8	K12	K16	K20	K24	K28	K32	K36	K40	K44	K48	K52	K56	K60
K1	K5	K9	K13	K17	K21	K25	K29	K33	K37	K41	K45	K49	K53	K57	K61
K2	K6	K10	K14	K18	K22	K26	K30	K34	K38	K42	K46	K50	K54	K58	K62
K3	K7	K11	K15	K19	K23	K27	K31	K35	K39	K43	K47	K51	K55	K59	K63

0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63

Figure 4: Add Round Key

C. Decryption algorithm

The decryption algorithm works exactly in the reverse order of the encryption process. The cipher text is taken as an input along with the key and the original text is outputted by the decryption algorithm. The algorithm is implemented in MATLAB and .Net Framework. Code optimization is done further to reduce the processing speed. Different text files with different size are given as an output.

V. REAL TIME APPLICATION SECURITY

The proposed system implements another approach to apply the security for real time applications such as voice data. The following algorithm is sampled with different voice file formats. MATLAB is used to sample the data.

1. Read the existing voice data file or record the data using Windows sound recorder and store the file in Wav format.

2. The audio file is sampled in MATLAB using the wavread( ) function which gives samples ranging from 1 to -1.
3. The samples is converted into array of integer data from 0 to 255 by adding one to each value and then multiply by 128.
4. Transfer the data to text file and store it in hard disk.
5. The data file is given as input to SF Block to encrypt the data and convert back to Wav format using the function wavwrite ( ).
6. The Decryption algorithm is used to get back the original data.

Spectrogram is generated for wav file before and after the encryption process. It was identified that original file and encrypted file had reasonable frequency change.

VI. Performance Evaluation

Performance is one of the vital components of any encryption algorithm. The proposed algorithm is implemented with performance in mind. This section gives detailed description about the simulation environment which is used to evaluate the performance of proposed algorithms. It also describes the system components that are used in the experiment. SF Block Cipher is implemented in two languages namely MATLAB and .NET Framework. AES algorithm and Blowfish algorithm was also implemented to evaluate the performances. AES uses the managed wrappers that are available in the System.Security.Cryptography Name Space. Blowfish is implemented using .Net framework.

A. Experimental Criteria:

Several performance metrics are used to evaluate the performance of the encryption algorithms such as Encryption time, Decryption time, CPU process time, and CPU clock cycles and Battery[Matin el at.,2009]. Encryption time is the total time taken to produce a cipher text from plain text. The calculated encryption time is then used to calculate the throughput of the encrypted algorithm. It gives the rate of encryption. The throughput of the encryption scheme is calculated as the total encrypted plaintext in bytes divided by the encryption time. Decryption time is the total time taken to produce the plain text from plain text. The calculated decryption time is then used to calculate the throughput of the decrypted algorithm. It gives the rate of decryption. The throughput of the decryption scheme is calculated as the total decrypted plaintext in bytes divided by the decryption time. The CPU process time is the time that is required to a CPU is dedicated only to the particular process of calculations. It reflects the load of the CPU. The CPU clock cycles are a metric, reflecting the energy consumption of the CPU while performing on encryption operations. Each cycle of CPU will consume a minute amount of energy

B. Experimental Procedures:

Several experimental procedures are used such as different encoding techniques for encryption, different packet sizes of data, different data types and different key sizes. In the case of encoding two types are used such as Base64 encoding and hexadecimal encoding. Packet size range from 0.5 MB to 20MB is used. Different data types such as text or document and images are used for each selected algorithms. Different key sizes are employed to trace the performance of the selected algorithms specifically power consumption. The formula to calculate the average encryption time is given in the equation (3)

$$AvgTime = \frac{1}{Nb}\sum_{i=1}^{Nb} \frac{Mi}{ti}(Kb/s)$$

Where
AvgTime = Average Data Rate (Kb/s)
Nb = Number of Messages
Mi=Message Size (Kb)
Ti=Time taken to Encrypt Message Mi
Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as in equation (4).

$$Throughput = \frac{Tp}{Et}$$

VII Results

The experimental procedures that are discussed above are used to evaluate the proposed algorithm. Different files sizes are used to test the performance of the algorithm. MATLAB is used to generate the graphs that show the encryption time and the throughput. Table I provides the encryption time for the proposed algorithm and Table II provides the decryption time for the proposed algorithm

Table 1: Encryption Time Data of SF Block Cipher

Input Size(Kb)	Time(Milliseconds)		
	AES	Blowfish	SF
49	59	36	56
59	39	36	38
100	94	61	90
247	121	90	112
321	167	134	164
694	234	256	210
899	254	256	258
963	213	187	208
5345	1324	1376	1237
7310	1432	1543	1366

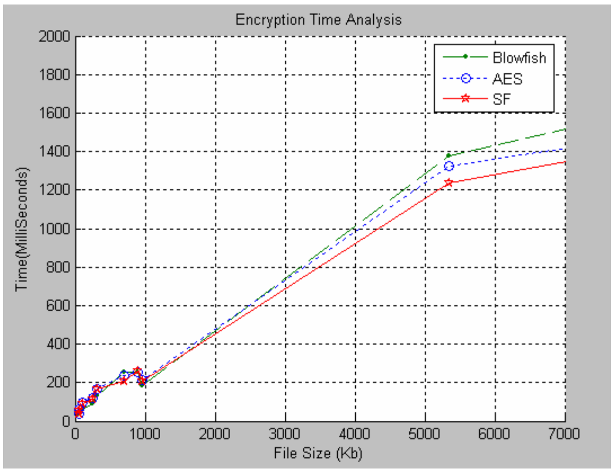


Figure 5: Encryption Time Analysis

TABLE 1I: DECRYPTION TIME DATA OF SF BLOCK CIPHER

Input Size(Kb)	Time(Milliseconds)		
	AES	Blowfish	SF
49	65	38	61
59	45	39	43
100	89	71	79
247	120	145	112
321	167	234	168
694	243	256	212
899	223	252	259
963	243	342	206
5345	1224	1371	1216
7310	1435	1443	1363

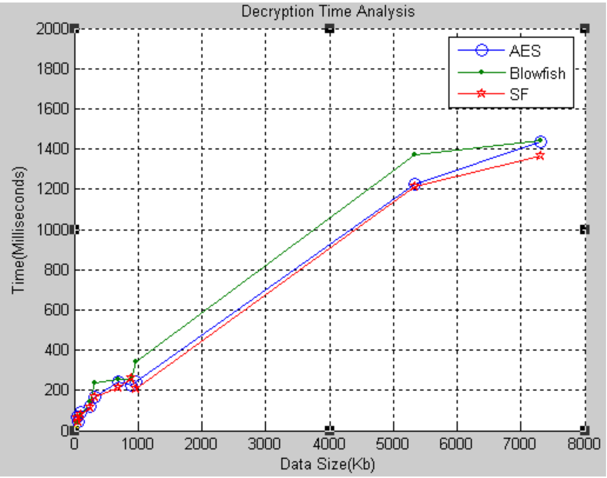


Figure 6: Decryption Time Analysis

Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated by dividing the total plaintext in Megabytes encrypted on the total encryption time for each algorithm in. As the throughput value is increased, the power consumption of the encryption technique is decreased. Table 3 shows the average time and throughput of SF Encryption algorithm and Table 4 shows the average time and throughput of SF Decryption algorithm.

TABLE III: THROUGHPUT ANALYSIS (ENCRYPTION)

Input Size(Kb)	Time(Seconds)		
	AES	Blowfish	SF
49	65	36	61
59	45	36	43
100	89	61	79
247	120	90	112
321	167	134	168
694	243	256	212
899	223	256	259
963	243	187	206
5345	1224	1376	1216
7310	1435	1543	1363
Average	388	395	377
Throughput	4.26	4.11	4.27

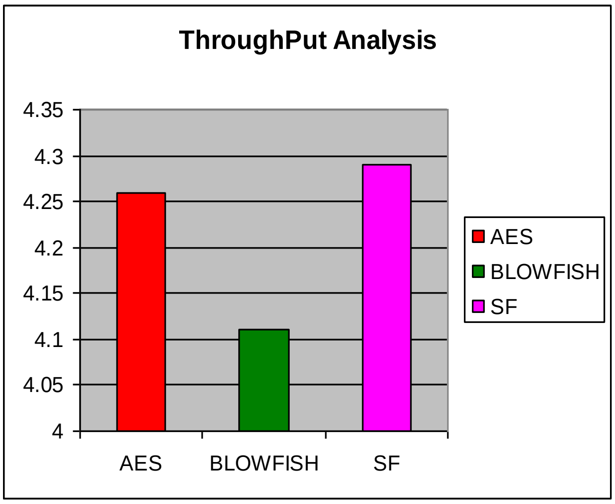


Figure 7: Throughput Analysis (Encryption)

TABLE 1V: THROUGHPUT ANALYSIS (DECRYPTION)

Input Size(Kb)	Time(Seconds)		
	AES	Blowfish	SF
49	59	36	56
59	39	36	38
100	94	61	90
247	121	90	112
321	167	134	164
694	234	256	210
899	254	256	258
963	213	187	208
5345	1324	1376	1237
7310	1432	1543	1366
Average	386	395	374
THROUGH put	4.29	3.34	4.59

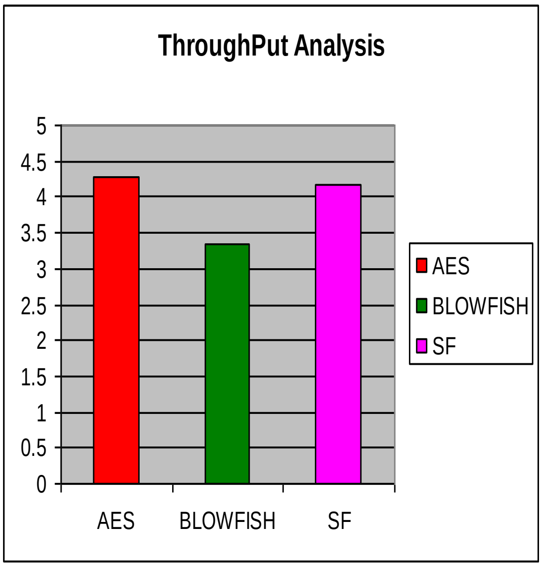
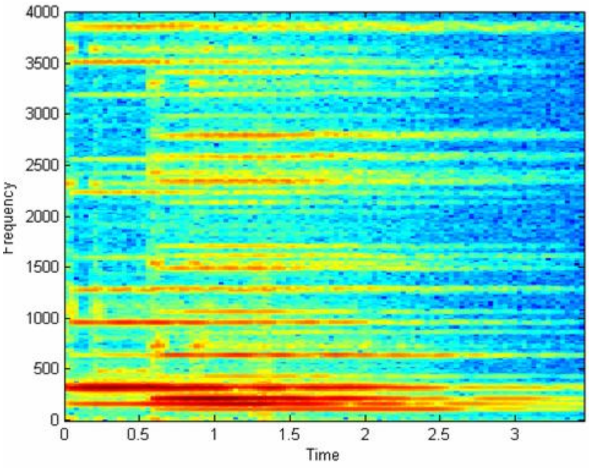


Figure 8: Decryption Time Analysis



From the analysis, it shows that the proposed Block cipher has better performance over other algorithms such as AES and Blowfish. Power consumption of SF Block is also good when compare to AES and Blowfish. But in the proposed architecture SF Block cipher is used along with other two algorithms namely Elgamal and SHA. So there is a slight overhead in terms of processing. So code optimization is done here to improve the performance. After applying the code optimization techniques, there is a huge improvement in terms of performance. Cryptanalysis is carried out in the encrypted file. It was found that the encrypted file with this algorithm is difficult to break.

Frequency analysis of Voice data after encryption using SF Block cipher: The analysis shows that there is a major change in the frequency of the audio signal after the encryption using the proposed model. It indicates that the proposed algorithm is highly secure and difficult to break the key. It is also identified that the algorithm best suited for real time applications such as video conferencing data and Voice over IP applications.

VIII. Conclusion

In this paper, various cryptographic algorithms are reviewed. From the literature review, it is identified that the block cipher with 128 bit keys and 128 bit block size will not be suitable for IT systems and banking sectors due to the advances in the computing technologies and cryptanalysis techniques. So in order to overcome this problem, we proposes a new block cipher named SF Block cipher which is capable of encrypting blocks of 512 bit size with the key size of 512 bit. Cryptanalysis is carried out in the encrypted file. It was found that the encrypted file with this algorithm is difficult to break. Performance is also analyzed for the proposed algorithm. Simulation results shows that the SF Block cipher has better performance over other algorithms. In this work, Key expansion algorithm of AES is used to generate the round keys. In future, a new key expansion algorithm will be proposed for this SF Block cipher.

References

[1] Alaa, T., A.A. Zaidan and B.B. Zaidan, 2009. New framework for high secure data hidden in the MPEG using AES encryption algorithm. Int. J. Comput. Electr. Eng., 1: 566-571.

[2] Alanazi, H.O., B.B Zaidan, A.A. Zaidan, A.H. Jalab, M. Shabbir and Y. Al-Nabhani, 2010. New comparative study between DES, 3DES and AES within nine factors. J. Comput., 2: 152-157.

[3] Behrouz A. Forouzan, 2005. TCP/IP Protocol Suite 3re Edn Tata McGraw Hill, pp: 30-46

[4] Bradner. S., 2006.The End-to-End Security IEEE Security & Privacy, 12:76-79.

[5] Colitti, L., Battista, G.D and Patrignani, M,2009.IPv6-in-IPv4 tunnel discovery: methods and experimental results. IEEE Transactions on Network and Service Management

[6] Dewu Xu Wei Chen., 2010.3G communication encryption algorithm based on ECC-ElGamal. International conference on signal processing systems, pp 47-54.

[7] Douligeris.C and Serpanos D., 2007. IP Security (IPSec). IEEE Book: Network Security: Current Status and Future Directions, 65 – 82

[8] Feistel, H., 1973. Cryptography and computer privacy. Sci. Am., 228: 15-23.

[9] Heng Yin Haining Wang., 2007.Building an Application-Aware IPsec Policy System. IEEE/ACM Trans. Networking, 15: 1502 – 1513

[10] Haraty, R.A., El-Kassar, A.N.and Shebaro, B.M., 2006. A Comparative Study of Elgamal Based Digital Signature Algorithms. IEEE Automation congress, 1-7.

[11] Jian, S., J. Sun, Z. Wang and Y. Dai, 2004. A fast video encryption scheme based-on chaos. Proceeding of the 8th IEEE International Conference on Control, Automation, Robotics and Vision, December 6-9, USA., pp: 126-131.

[12] Jiang, J., 1996. Pipeline algorithms of RSA data encryption and data compression. Proceeding of IEEE International Conference on Communication Technology, May 5-7, Beijing, China, pp: 1088-1091.

[13] Macedo, D.F., dos Santos, A.L and Pujolle, G. 2008. From TCP/IP to convergent networks: challenges and taxonomy IEEE Communication Survey and tutorial 10:40-55

[14] Mathis, M, 2009. Reflections on the TCP Macroscopic Model, ComputerCommunication Review

[15] Matin, M.A. Hossain, M.M. Islam, M.F and Islam, M.N,2009. Performance evaluation of symmetric encryption algorithm in MANET and WLAN.IEEE International conference for Technical Postgraduates, pp: 1-4.

[16] Mohammad Al-Jarrah and Abdel-Karim R. Tamimi., 2007.A Thin Security Layer Protocol over IP Protocol on TCP/IP Suite for Security Enhancemen. IEEE Conference in Innovations in Information Technology,1-5

[17] Philip M. Miller., 2009.TCP/IP The ultimate Protocol Guide Data delivery and Routing. Brown walker press

[18] Rabah, K., 2004. Data security and cryptographic techniques: A review. Inform. Technol. J., 3: 106-132.

[19] Rabah, K., 2005. Secure implementing message digest. Authentication and Digital Signature

[20] Wanzhong Sun, Hongpeng Guo, Huilei He and Zibin Dai,2007. Design and optimized implementation of the SHA-2(256, 384, 512) hash algorithms.IEEE Conference on ASIC Processing, 858-861.

[21] Zaidan, B.B., A.A. Zaidan, A.K. Al-Frajat and H.A. Jalab, 2010. On the differences between hiding information and cryptography techniques: An overview. J. Applied Sci., 10: 1650-1655.

[22] Zaidan, B.B., A.A. Zaidan, A.K. Al-Frajat and H.A. Jalab, 2010. An Overview: Theoretical and Mathematical Perspectives for Advance Encryption Standard/Rijndael. Journal of Applied Sciences, 10: 2161-2167.

[23] W.Stallings, "Cryptography and Network Security 4th Ed," Prentice Hall , 2005,PP. 58-309 .

[24] Masashi Une and Masayuki Kanda, 2007. Year 2010 Issues on Cryptographic Algorithms, Monetary And Economic Studies, Pp 129-164

M. Anand Kumar received the B.Sc. and M.Sc. degrees in Computer Science from Bharathiar University, Coimbatore, India, in 2001 and 2003 respectively. He is Lecturer at the Department of Information Technology, Karpagam University, India. He is pursuing his doctoral degree at Karpagam University. His area of research includes network security and information security. He has presented fifteen papers in national conferences and four papers in international conferences. He has published eight papers in international journals

Dr.S.Karthikeyan presently working as Assistant Professor, College of Applied Sciences, Oman and previously he was a Senior Lecturer at Caledonian College of Engineering, Oman. He was a Professor & Director at Karpagam University, School of Computer Science and Applications, Coimbatore. He has total of 14 years of teaching and research experience. Dr.Karthikeyan completed his PhD at Alagappa University, Karaikudi, India in the area of Network Security, Computer Science and Engineering by Feb 2008. He has 32 research papers and guiding 11 PhD research scholars from various universities in India and he has also guided 19 M.Phil students. He is Chief and guest editor of various national and international journals. He has chaired many conference sessions and served as Technical Committee member of various boards at various colleges, universities and conferences.