International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 1, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

New Encryption Standard and Particular Case of Blowfish Algorithm

Chandra Sekhar Maganty
Ph.D. Research Scholar
CMJ University, Meghalaya, India
chandra_maganty@yahoo.co.in

K. Sai Prasanthi
Ph.D. Research Scholar
CMJ University, Meghalaya, India
prash_sru@yahoo.co.in

Abstract **- Encryption algorithm plays a crucial role in information security which guarantees the recent growing internet and network applications. They are used to secure the data in wireless networks against malicious attacks but securing data also consumes resources such as C.P.U time, Memory, battery power, encryption time etc. Blowfish, a new secret-key block cipher, is proposed. It is a Feistel network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be any length up to 448 bits. Looking at the current situation, our ambition is to tighten up the security factor that minimizes the effect of Cryptanalysis being done on Blowfish algorithm. Our new paradigm is having substantial amount of effort and an improvable aspect on security over the network. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors. This paper discusses the requirements for a standard encryption algorithm. Experimental results show that Blowfish encryption algorithm may be more suitable for wireless networks with secure data transmission. With reduced bits in plain text and key our aim is to make it useful for systems having minimum configuration.**

Keywords **- Block Cipher, Fiestel Network, Standard Encryption, Cryptanalysis.**

I. INTRODUCTION

Information is an asset that has a value like any other asset. As an asset, information needs to be secured from attacks. Now-a-days security becomes an essential feature in almost all area of communication. While sending a message to a person over an insecure channel such as internet we must provide confidentiality, integrity, authenticity and non-repudiation. These are the four major security aspects or goals. We have a number of encryption algorithms those can be broadly classified into two categories: *Symmetric/Private key encipherment* and *Asymmetric/ Public key encipherment* .

The difference between these two is that to communicate with *n* people *private key cryptography* requires (*n**(*n*-1))/2 number of keys whereas; *public key cryptography* requires only *n* number of key pairs (one private and one public key).

Asymmetric key algorithm uses two keys, one is used to encrypt the data and other is used to decrypt the data. Length of Key has an important place in Symmetric key encryption [17].

A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key – a word, number, or phrase – to encrypt the plaintext. The same plaintext encrypts to different cipher text with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key. Blowfish Algorithm is such a block algorithm designed in the year 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Blowfish Algorithm is a symmetric block cipher that can be used as a replacement for DES or IDEA.. Because it keeps a lot of secret information, including S-boxes, the algorithm is secure and easy to understand. In the original Blowfish algorithm the keys are expanded and should be pre computed before every encryption and decryption. But in Proposed Blowfish algorithm the keys are generated randomly and saved in the P- Array as discussed in Section 2.

Data that can be read and understood without any special measures is called Plain text or Clear text. The method of disguising plain text in such a way as to hide its substance is called Encryption. Encrypting plaintext results in unreadable gibberish called *Cipher Text.* You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting cipher text to its original plain text is called *Decryption* Figure below illustrates this process.

Electronic commerce and other forms of secure communications require adherence to four fundamental security principles that cryptography greatly enhances. These principles are Privacy, Authentication, Data Integrity, and Non - Repudiation.

Fig.1.

In Section II Blowfish Algorithm is discussed and the keys used in it and also the pictorial representation of the algorithm is shown. Section III shows the detailed process of the New Encryption Standard and Particular Case of Blowfish Algorithm. In Section IV, results of the tests performed are shown to prove the effectiveness of the algorithm.

II. LITERATURE REVIEW

Many Cryptographic Algorithms were proposed in last two decades. One of the important among them is GOST Algorithm.

2.1 Research Background

Diaa Salama et.al paper presents a performance evaluation of selected symmetric encryption algorithms on power consumption for wireless devices. The selected algorithms are AES, DES, RC6, Blowfish, RC2 and 3DES. There is insignificant difference between open key authentications and shared key authentication in ad hoc Wireless LAN connection with excellent signals. In case of poor signal it is found that, transmission time increased minimum by 70 % over open sheered authentication in ad hoc mod [12].

Simar Preet Singh et.al (2011) study reveals that Blowfish has better performance than other commonly used encryption algorithms. Since Blowfish has not any known security weak points so far, it can be considered as an excellent standard Encryption algorithm. AES showed poor performance results compared to other algorithms, since it requires more processing power [13].

M. Umaparvathi et.al (2010) discussed the comparison of the most commonly used symmetric encryption algorithms AES (Rijndael), DES, 3DES and Blowfish in terms of power consumption. A comparison has been conducted for those encryption algorithms at different data types like text, image, audio and video. Results showed that AES has a better performance than other common encryption algorithms used. Since AES has not any known security Weak points so far, this makes it an excellent candidate to be considered as a standard encryption algorithm. 3DES showed poor performance results compared to other algorithms since it requires more processing power [16].

2.2 Conventional Blowfish Algorithm

Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Since then it has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm [1]. Blowfish is unpatented and license-free, and is available free for all users.

For the same algorithm, encryption using longer key is hard to crypt analyze means more secure as compared to the one using shorter key. Asymmetric encryption techniques are almost one-thousand times slower than symmetric techniques as they require more computational processing power [11].

2.3 New Approach for modifying Blowfish Algorithm by using Multiple Key's

In this paper, cellular automata (CAs) are used to design a symmetric key cryptography system based on Bluefish algorithm, CAs are applied to generate a multiple pseudo-random numbers sequence (PNS) which is used during the encryption process. The quality of PNSs highly depends on the set of applied CA rules. This paper introduces a new method to enhance the performance of the Bluefish Algorithm. This is done by building a new structure for the 16 rounds in the original algorithm by replacing the OR operation with a new introduced operation.

This structure makes use of multiple secrete keys. The principle of Cellular Automata (CA) is used to generate these multiple keys in a simple and effective way. The proposed method provides high quality encryption, and the system is very resistant to attempts of breaking the cryptography key.

2.4 Approach of Conventional Blowfish

Blowfish is a 64-bit block cipher with a variable-length key. The algorithm consists of two parts: key expansion and data encryption. Key expansion converts a key of up to 448 bits into several sub key arrays totaling 4168 bytes. Data encryption consists of a simple function iterated 16 times. Each round consists of a key dependent permutation, and a key- and data-dependent substitution. All operations are additions and XORs on 32-bit words. The only additional operations are four indexed array data lookups per round.

Blowfish uses a large number of sub keys. These keys must be pre computed before any data encryption or decryption.

The P-array consists of 18 32-bit sub keys:

P_1, P_2,..., P_{18}

Four 32-bit S-boxes have 256 entries each:

*S*1,0, *S*1,1,..., *S*1,255

*S*2,0, *S*2,1,..., *S*2,255

*S*3,0, *S*3,1,..., *S*3,255

*S*4,0, *S*4,1,..., *S*4,255

The exact method used to calculate these sub keys will be described later.

The Fiestel Structure of Conventional Blowfish Algorithm and its working is explained below.

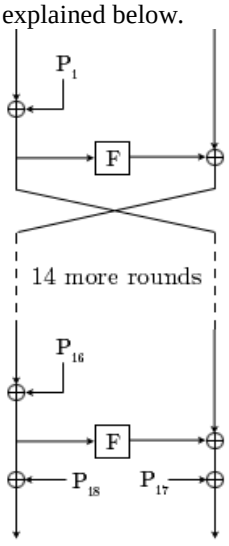

Fig.2.4.1 Fiestel Structure of Blowfish

Blowfish is a Feistel network consisting of 16 rounds. The input is a 64-bit data element, *x*. To encrypt:

Divide *x* into two 32-bit halves: X_L and X_R

For i = 1 to 16:

$X_L = X_L \oplus Pi$

$X_R = F(X_L) \oplus X_R$

Swap X_L and X_R

Swap X_L and X_R (Undo the last swap.)

$X_R = X_R \oplus P_{17}$

$X_L = X_L \oplus P_{18}$

Recombine X_L and X_R

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 1, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Recombining X_L and X_R, we obtain the Cipher Text of 64-bit.

The $F(X_L)$ is the function, and X_L is the input for the function in every round which is 32-bit.

Inside the function, the input X_L of 32-bit is divided into four equal halves of 8-bit each, and each 8-bit half is given to S-Box, and the output of every S-Box is again a 8-Bit. So, the four S-Boxes generate four 8-bit outputs. The four 8-bit outputs are added modulo 2^{32} and XORed to produce the final 32-bit output, which is the final output of the function.

The function $F(X_L)$ is represented as follows.

Function F:

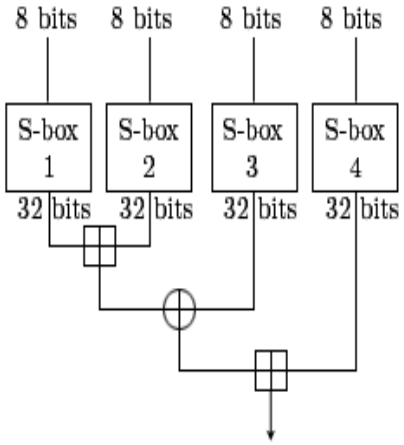

Figure 2.4.2 The Function F

The Function F is as follows:

F-function splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added modulo 2^{32} and XORed to produce the final 32-bit output.

Divide X_L into four eight-bit quarters and name it as a, b, c and d:

$$F(X_L) = ((S1, a + S2, b \bmod 2^{32}) \oplus S3, c) + S4, d \bmod 2^{32} \quad \text{---------- (1)}$$

Decryption is exactly the same as encryption, except that $P_1, P_2, \ldots, P_{18}$ are used in the reverse order.

This is the process of the Encryption and the Decryption takes place in the Conventional Blowfish Algorithm.

2.5 Security of Blowfish

Serge Vaudenay examined Blowfish with known S-boxes and *r* rounds; a differential attack can recover the P-array with 2^{8r+1} chosen plaintexts. For certain weak keys that generate bad S-boxes (the odds of getting them randomly are 1 in 214), the same attack requires only 2^{4r+1} chosen plaintexts to recover the P-array. With unknown S-boxes this attack can detect whether a weak key is being used, but cannot determine what it is (neither the S-boxes nor the P-array). This attack only works against reduced-round variants; it is completely ineffective against 16-round Blowfish. Of course, the discovery of weak keys is significant, even though they seem impossible to exploit. A weak key is one in which two entries for a given S-box are identical. There is no way to check for weak keys before doing the key expansion. If you are worried, you have to do the key expansion and check for identical S-box entries. I don't think this is necessary, though. I know of no successful cryptanalysis against Blowfish. To be safe, do not implement Blowfish with a reduced number of rounds.

III. New Encryption Standard and Particular Case of Blowfish Algorithm

3.1 Overview:

In the New Encryption Standard and Particular Case of Blowfish Algorithm, the Block size is reduced from 64-bit to 32-bit; hence the plain text per block will be of size 32-bit. The key size is 64-bit. The sub keys are each of 16-bit. It should be noted that once the 64-bit Key is generated, it should be stored such that it need not have to be generated upon every encryption. The number of rounds or iterations will be same as Conventional blowfish algorithm of 16 rounds. It consists of 4 S-boxes. Each S-box gets an input of 4-bits and produce an output of 16-bits.The structure we follow is the Fiestel Cipher structure. Our algorithm has certain considerations as follows-

The plain text written is first converted into ASCII values thereby converted into binary numbers.

Now after the conversion, the plain text size is of 32-bit. The total number of rounds for this algorithm to complete is 16. The key size is 64-bit. The sub keys are each of 16-bit. It should be noted that once these sub keys are generated, they can be stored such that they do not have to be generated upon every encryption

3.2 Generating Sub Keys:

The P-array consists of 18 entries and it consists of 18 16-bit sub keys. The algorithm generates 64-bit Key using random function. The generated 64-bit Key is divided into 4 16-bit keys. These 4 keys are used for first four rounds. After first four rounds the original 64-bit key is given left circular shift of 7-bit, after eighth round the original key is given left circular shift of 9-bit, similarly after twelveth round the original key is given left circular shift of 11-bit, after sixteenth round the original key is given left circular shift of 13-bit.

This provides more security because of non-repetition of the key and also it is not necessary to generate 18 i6-bit sub keys using some random function. The design of the function to generate random number is specific to the organization using this algorithm which is generally not disclosed. Thus the algorithm provides much security and is rich in privacy.

In the proposed Blowfish, only the plain text size and key size is reduced, so the block size will be reduced so it will be an advantage as in general people are focusing on larger block size like AES and Two fish algorithms etc. As the block size is small than conventional blowfish, the encryption and decryption can be little bit easier.

We can generate the Key using any function or by generating randomly.

The S-box generation can be done using the Following operations:

1. XOR
2. ADDITION
3. MULTIPLICATION
4. LOOK UP TABLES

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 1, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

In the proposed blowfish we are considering the XOR operation. It should be noted that once these sub keys and S-boxes are generated, they can be stored such that they do not have to be generated upon every encryption.

3.3 Approach

In the proposed blowfish, the operations are same as the conventional Blowfish, but the only operation changing is XOR operation instead of table lookups in the S-box bits generation. The remaining operations are same as in the conventional blowfish. The operations are simple XOR's on 16-bit words and simple addition operations on S-boxes output bits. The design of proposed Blowfish algorithm is

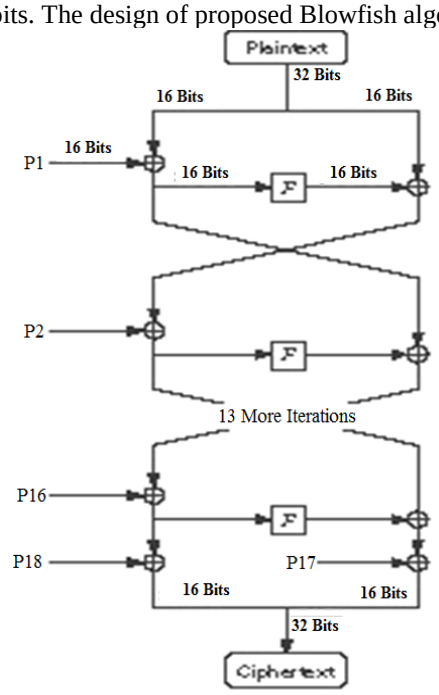

Fig.3.3.1 Fiestel Structure of Proposed Blowfish Algorithm

Particular Case of Blowfish Algorithm is a Feistel network consisting of 16 rounds. The input is a 32-bit data element, *x*. To encrypt:

Divide *x* into two 16-bit halves: X_L and X_R

For i = 1 to 16:

$X_L = X_L \oplus Pi$

$X_R = F(X_L) \oplus X_R$

Swap X_L and X_R

Swap X_L and X_R (Undo the last swap.)

$X_R = X_R \oplus P_{17}$

$X_L = X_L \oplus P_{18}$

Recombine X_L and X_R

Recombining X_L and X_R, we obtain the Cipher Text of 32-bit.

The $F(X_L)$ is the function, and X_L is the input for the function in every round which is 16-bit.

Inside the function, the input X_L of 16-bit is divided into four equal halves of 4-bit each, and each 4-bit half is given to S-Box, and the output of every S-Box is again a 4-Bit. So, the four S-Boxes generate four 4-bit outputs. The four 4-bit outputs are added modulo 2^{16} and XORed to produce the final 16-bit output, which is the final output of the function.

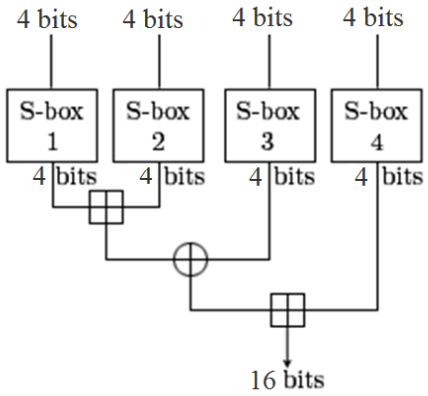

Fig.3.3.2 The Function F in Proposed Blowfish

The Function F is as follows:

F-function splits the 16-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added modulo 2^{16} and XORed to produce the final 16-bit output.

Divide X_L into four eight-bit quarters and name it as a, b, c and d:

$F(X_L) = ((S1, a + S2, b \bmod 2^{16}) \oplus S3, c) + S4, d \bmod 2^{16}$

---------- (1)

Decryption is exactly the same as encryption, except that $P_1, P_2, \ldots, P_{18}$ are used in the reverse order.

This is the process of the Encryption and the Decryption takes place in the Conventional Blowfish Algorithm.

S-Box Operation:

In Particular Case of Blowfish Algorithm, we said that it consists of four S-Boxes. For every S-Box we are giving 4-bit input and every S-Box will be generating 4-bit output. Generally S-Boxes deals with Look up tables, but in Particular Case of Blowfish Algorithm, instead of Look up tables, we are going with generating 4-bit random number and XORing with the input given to each S-Box, so that every S-Box will be generating 4-bit output.

Let us assume the output of X_L is:

0010 0100 0010 1100 …….. (1)

Dividing this 16-bit into four equal parts –

Part 1	Part2	Part3	Part4
0010	0100	0010	1100

Each part is given as input to S-Box

After generating the random number

Suppose

1111 1001 0111 1010…… (2)

Dividing it into four equal parts—

RaNum1	RaNum2	RaNum3	RanNum4
1111	1001	0111	1010

Now XOR (1) and (2) i.e. part1 with RaNum1, part2 with RaNum2 and this process for all the four parts.

Finally the outputs are added modulo 2^{16} and XORed to produce the final 16-bit output.

This provides more security because of generating the sub keys randomly instead of generating them with some predefined permutations.

The design of the function to generate random number is specific to the organization using this algorithm which is generally not disclosed. Thus the algorithm provides much security and is rich in privacy.

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 1, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

IV. RESULTS

Plain Text: We have considered the Plain text as "**3923780913"**. This plain text is obtained from the relevant ASCII values. As discussed the plain text is of 32-bit size and sub keys size is of 16-bit.

Considering the following as the plain text
Equivalent Binary Data of Plain Text:
11101001111000000010010100110001
Key: (64 Bit):
11100011000010101111101110001001111100001000011
10100010101100011

Table 1: Rounds, Keys, Left Half and Right Half of Plain Text after Every Round

ROUNDS and KEYS		PLAIN TEXT 11101001111000000010010100110001	
ROUNDS	SUB-KEYS	X_L=1110100111100000	X_R=0010010100110001
R_1	P_1=1110001100001010	0000101011101010	0010100001001100
R_2	P_2=1111101110001001	1101001111000101	1011100001010001
R_3	P_3=1111000010000111	0100100011010110	0001011100000101
R_4	P_4=0100010101100011	0101001001100110	0101101011111010
R_5	P_5=1000010101111101	1101111110000111	0101010101010100
R_6	P_6=1100010011111000	1001000110101100	0010100001001100
R_7	P_7=0100001110100010	0110101111101110	1111100001101100
R_8	P_8=1011000111110001	0100100110011101	0101001100000101
R_9	P_9=0001010111110111	0100011011110010	1101111111000110
R_{10}	P_{10}=0001001111100001	1100110000100111	1101111111000110
R_{11}	P_{11}=0000111010001010	1101000101001100	0100100110111001
R_{12}	P_{12}=1100011111000110	1000111001111111	0101101011111010
R_{13}	P_{13}=0101011111011100	0000110100100110	0101101011111010
R_{14}	P_{14}=0100111110000100	0001010101111110	0001011100000101
R_{15}	P_{15}=0011101000101011	0010110100101110	1011100001010001
R_{16}	P_{16}=0001111100011000	1010011101001001	1011100001010001
	P_{17}=0101111101110001		1110011100100000
	P_{18}=0011111000010000	1000011001000001	

Observing Table 1 we see that the sub keys are non-repetitive which is the most essential factor of this algorithm. The Conventional Blowfish does not have this; it generates the sub keys with some pre existing permutations which give existence for a cryptanalytic attack. Following the proposed one the results obtained are as follows-

By using the approach of proposed algorithm with the corresponding sub key X_L and X_R are generated for each round. After the completion of 16th round, X_L is XORed with P_{18} and X_R is XORed with P_{17}. The final X_L and X_R are the results.

Finally now combining X_L and X_R, we obtain the Cipher Text.

Cipher Text: 10000110010000011110011100100000

Equivalent Binary Data of Cipher Text is obtained; now transforming this binary data into ASCII is done.

So, for the plain text **"3923780913"**, after encrypting it we got the Cipher Text **"2252465952"**.

V. Conclusion and Future Scope

In the New Encryption Standard and Particular Case of Blowfish Algorithm, we have implemented the algorithm with improved Security than the conventional Blowfish algorithm as in the Conventional Blowfish algorithm, the 18 Sub-Keys are generated by pre existing permutations or functions, and there is a chance of repetition of sub keys, whereas in the New Encryption Standard of Blowfish Algorithm, we need not generate all the 18 sub keys, instead of that, we generate only one Key of 64-bit, for every four rounds the Original Key is being changing with Left Circular Shift of 5, 7, 9, 11 and 13 bits respectively for every four rounds. So, with the generated Key of 64-bit will generating 18 16-bit sub keys without the repetition of the key. The 64-bit Key is generated randomly using function and due to left circular shift of Key after every four rounds, there is no chance of repetition of sub keys. In the New Encryption Standard of Blowfish algorithm, the block size of Plain Text is being reduced from 64 bit to 32 bit as of Conventional Algorithm, as generally people will be looking to break the Algorithms of larger block size. As the result in reducing the block size, we have also reduced the Key size to 16 bit from 32 bit Key of Conventional Algorithm. The future scope is to implement the same algorithm with including S-Boxes with Look Up Tables in the function of the proposed Algorithm and also to increase the block size from 32-bit to 40, 64,128 bits with improving security on larger block size in the proposed algorithm.

References

[1] B. Schneier, Applied Cryptography. John Wiley and Sons, 1996. ISBN 0-471-11709-9.

[2] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd edition, John Wiley & Sons, Inc., 1996.

[3] Alan G. Konheim, "Computer Security And Cryptography ",2007 , by John Wiley & Sons, Inc.

[4] Alfred J.M., Paul V. C. and Scott A. V., "Handbook of Applied Cryptography", Fifth Addition, 2001.

[5] B. Schneier, Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish) Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, 1994, pp. 191-204.

[6] Coppersmith, Don. (1994). "The data encryption standard (DES) and its strength against attacks". IBM Journal of Research and Development, 38(3), 243–250.

[7] National Institute of Standards and Technology, (1979). "FIPS-46: Data Encryption Standard (DES)." Revised as FIPS 46-1:1988, FIPS 46-2:1993, FIPS 46-3:1999, available at http://csrc.nist7aznml;'.gov/publications/fips/ fips46-3/fips46-3.pdf

[8] Hala Bahjat Abdul Wahab1 , Abdul Monem S. Rahma, 'Proposed New Quantum Cryptography System Using Quantum Description techniques for Generated Curves", The 2009 International conference on security and management, SAM2009, July 13-16 2009, Las Vegas, USA, SAM 2009.

[9] Henk C.A. van Tilborg, Eindhoven, "Encyclopedia Of Cryptography And Security", 2005, Springer Science+Business Media, Inc.

[10] Afaf M. Ali Al-Neaimi, Rehab F. Hassan, "New Approach for Modifying Blowfish Algorithm by Using Multiple Keys", IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.3, March 2011

[11] Hardjono, "Security in Wireless LANs and MANs", Architect House Publishers, 2005.

[12] Diaa Salama, Hatem Abdual Kader and Mohiy Hadhoud (2011), "Studying the Effects of Most Common Encryption Algorithms", International Arab Journal of e- Technology, Vol. 2, No. 1, January 2011, pp 1-10.

[13] SimarPreet Singh, and Raman Maini (2011), "Comparison of Data Encryption Algorithms", International Journal of Computer Science and Communication Vol. 2, No. 1, January-June 2011, pp. 125 – 127.

[14] A.Rathika, Parvathy Nair and Parvathy Nair (2011), "A High Throughput Algorithm for Data Encryption" International Journal of Computer Applications (0975 – 8887) Volume 13, No.5, January 2011 pp 13-16.

[15] Lavanya P and M Rajashekhara Babu (2011), "Performance Analysis of Montgomery Multiplication Algorithm for Multi-core Systems Using Concurrent Java", Journal of Advances in Applied Science Research, 2011, 2 (3), pp 567-573.

[16] M.Umaparvathi, Dr.Dharmishtan and K Varughese (2010), "Evaluation of Symmetric Encryption Algorithms for MANETs", Proceedings of 2010 IEEE International conference on Computational Intelligence and Computing Research (ICCIC-2010), 28-29 Dec. 2010, pp 1-3.

[17] William Stalling, "Cryptography and Network Security Principles and Practice 5th Edition", Pearson.

Author's Profile

K. Sai Prasanthi
received B.Tech. and M.Tech. Degrees in Computer Science & Information Technology and Computer Science under the University of JNTU, Hyderabad & Berhampur University, Berhampur in the year 2007 and 2011 respectively. She is currently pursuing her PhD in CMJ University, Meghalaya. She is an Assistant Professor in the department of Computer Science & Engineering from 2010 June, in GIET Gunupur; she was an Assistant Professor in the Department of CSE from 2007 to 2010, in AIET, Andhra Pradesh. Her research interests include Network Security and Cryptography, Image Processing, Real Time Systems.

Chandra Sekhar Maganty
received B.Tech. and M.Tech. Degrees in Computer Science & Information Technology and Computer Science under the University of JNTU, Hyderabad & Berhampur University, Berhampur in the year 2006 and 2011 respectively. He is currently pursuing his PhD in CMJ University, Meghalaya. He is an Assistant Professor in the department of Computer Science & Engineering from 2010 June, in GIET Gunupur; he was an Assistant Professor in the Department of CSE from 2006 to 2010, in AIET, Andhra Pradesh. His research interests include Network Security and Cryptography, Image Processing, Real Time Systems.