

Modified Feistel Cipher Involving Interlacing and Decomposition

K.Anup Kumar¹ and V.U.K. Sastry²

¹Associate Professor, Department of Computer Science and Engineering, SNIST,
Hyderabad, Andhra Pradesh, India.
¹k_anupkumar@yahoo.com

²Dean R & D, Department of Computer Science and Engineering, SNIST,
Hyderabad, Andhra Pradesh, India.
²vuk_sastry@rediffmail.com

Abstract: In this paper, we have discussed the generation of large block cipher of 256 bit by using the modified feistel structure involving basic concepts of interlacing, decomposition and key based random permutations. In each round, we perform decomposition before encryption and interlacing after encryption. The key based random permutations and substitutions used in this process are similar to the one we already published in our previous paper. The cryptanalysis carried out in this paper, indicates that the cipher cannot be broken by any cryptanalytic attack due to the non linearity induced by the interlacing, decomposition and key based random permutations.

Keywords: Encryption, Decryption, Plaintext, Cipher text, Key, Interlacing, Decomposition etc.

1. Introduction

In the survey of literature of cryptography, Feistel structure has a predominant role in generating the block cipher of required size. Here, the bits of the plaintext undergo a series of diffusion and confusion transformations involving permutations, substitutions. The classical feistel structure involves a round function and the number of rounds which provides good strength to the cipher is sixteen.

In this paper, we have developed a block cipher of 256 bit, using 16 rounds of classical feistel structure. In the process of encryption and decryption, we have used the function 'F' in each round same as our conventional feistel structure with key based random permutations and substitutions published in our previous paper, see reference [6]. To get proper mixing of bits between two consecutive rounds; to introduce the non linearity and counter attack the cryptanalysis, we have used the concepts of interlacing and decomposition. Our interest is to develop a block cipher using feistel network which cannot be broken by any cryptanalytic attack.

In section 2 of this paper, we introduce the process of interlacing and decomposition in feistel network followed by the process of interlacing and decomposition demonstrated in figure. In section 3, we discuss the development of cipher and we present the algorithms for encryption, decryption, Let 'Cⁱ' be the 256 bit cipher obtained after interlacing the ciphers c^{m+1}₁, c^{m+1}₂, c^{m+1}₃, c^{m+1}₄. Here 'i' indicates the round interlacing and decomposition in

section 4. We have illustrated the cipher in section 5 and investigated the cryptanalytic attack on cipher in section 6. In section 6.3, we have discussed the avalanche effect which is followed by the conclusion in section 7 and reference in section 8.

2. Interlacing and Decomposition

Let us illustrate the process of decomposition first. Let 'P' be the plaintext of length 256 bit. Let us divide this plaintext of 256 bit block into four small blocks of 64 bits each.

Let C⁰ = P be the initial plaintext. Thus we get, B⁰₁, B⁰₂, B⁰₃, B⁰₄ as 64 bits blocks by placing the first 64 bits of 'C⁰' in 'B⁰₁' and the next 64 bits of 'C⁰' in 'B⁰₂' and so on.

Hence,

C^k = Σ B^k_{i,j}. Such that, i = 1 to 4 and j = 1 to 64. k = 0 to 16; Where, k = 0 indicates initial plaintext, k = m indicates cipher text after mth round and Σ indicates concatenation of bits.

Let C⁰ = { C⁰₁, C⁰₂, C⁰₃, ..., C⁰₂₅₆ }. Then, B^m_i = Σ C^m_{j+k}. Where, i = 1 to 4, j = 1 to 64 and k = 64*(i - 1). therefore,

$$B^m_1 = \{ C^m_1, C^m_2, C^m_3, \dots, C^m_{64} \} \quad (2.1)$$

$$B^m_2 = \{ C^m_{65}, C^m_{66}, C^m_{67}, \dots, C^m_{128} \} \quad (2.2)$$

$$B^m_3 = \{ C^m_{129}, C^m_{130}, C^m_{131}, \dots, C^m_{192} \} \quad (2.3)$$

$$B^m_4 = \{ C^m_{193}, C^m_{194}, C^m_{195}, \dots, C^m_{256} \} \quad (2.4)$$

We perform decomposition before encryption. So that, a large block of 256 bit is divided into a small block of 64 bit. Hence encryption of these small blocks can be done in parallel and faster. Moreover, decomposition allows us to introduce enough confusion in a large block cipher due to which the desired avalanche effect is maintained. See (6.3).

Now let us illustrate the process of interlacing.

We perform interlacing after encryption is performed on small blocks B^m₁, B^m₂, B^m₃, B^m₄.

Let c^{m+1}₁, c^{m+1}₂, c^{m+1}₃, c^{m+1}₄ be the corresponding ciphers obtained after encryption.

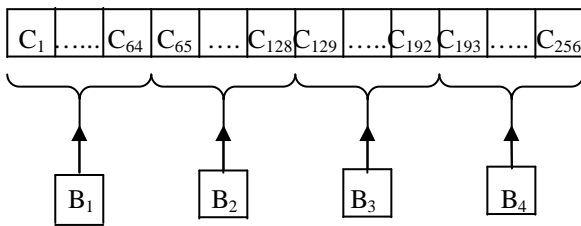
Let 'Cⁱ' be the 256 bit cipher obtained after interlacing the ciphers c^{m+1}₁, c^{m+1}₂, c^{m+1}₃, c^{m+1}₄. Here 'i' indicates the round after which interlacing is performed and i=m+1. In the process of interlacing, we take the first bit of 'c^{m+1}₁' and place it as the first bit of Cⁱ, next we take the first bit of

' c^{m+1}_2 ' and place it as the second bit of C^i , and similarly the first bit of ' c^{m+1}_3 ' and ' c^{m+1}_4 ' are placed as the third and fourth bit of C^i . This process is continued till all the bits of $c^{m+1}_1, c^{m+1}_2, c^{m+1}_3, c^{m+1}_4$ are combined into C^i . Therefore

$$C^i = \{ c_{1,1}, c_{2,1}, c_{3,1}, c_{4,1}, c_{1,2}, c_{2,2}, c_{3,2}, c_{4,2}, \dots, c_{1,64}, c_{2,64}, c_{3,64}, c_{4,64} \} \quad (2.5)$$

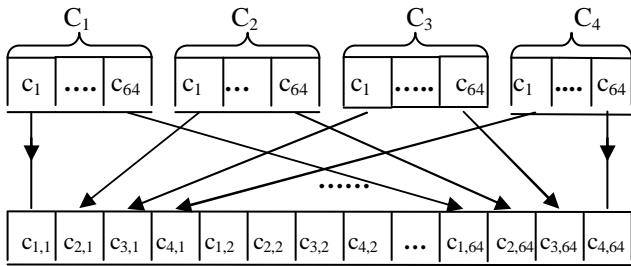
Thus, The process of interlacing allows us to mix the bits thoroughly before beginning the next round. Interlacing and decomposition enables us in performing variable permutations and substitutions on bits in each round. The following figures explain how interlacing and decomposition are used.

Decomposition



64 bit blocks B_1, B_2, B_3, B_4 obtained after Decomposition

Interlacing



Cipher text C^i of 256 bits after Interlacing.

3. Development of Cipher

Let us consider a block of plaintext 'P' consisting of 32 characters. By using the EBCDIC code, each character can be represented in terms of 8 bits.

Then the entire plaintext of 32 characters yields us a block containing 256 bits.

Let this initial plaintext be represented as C^0 .

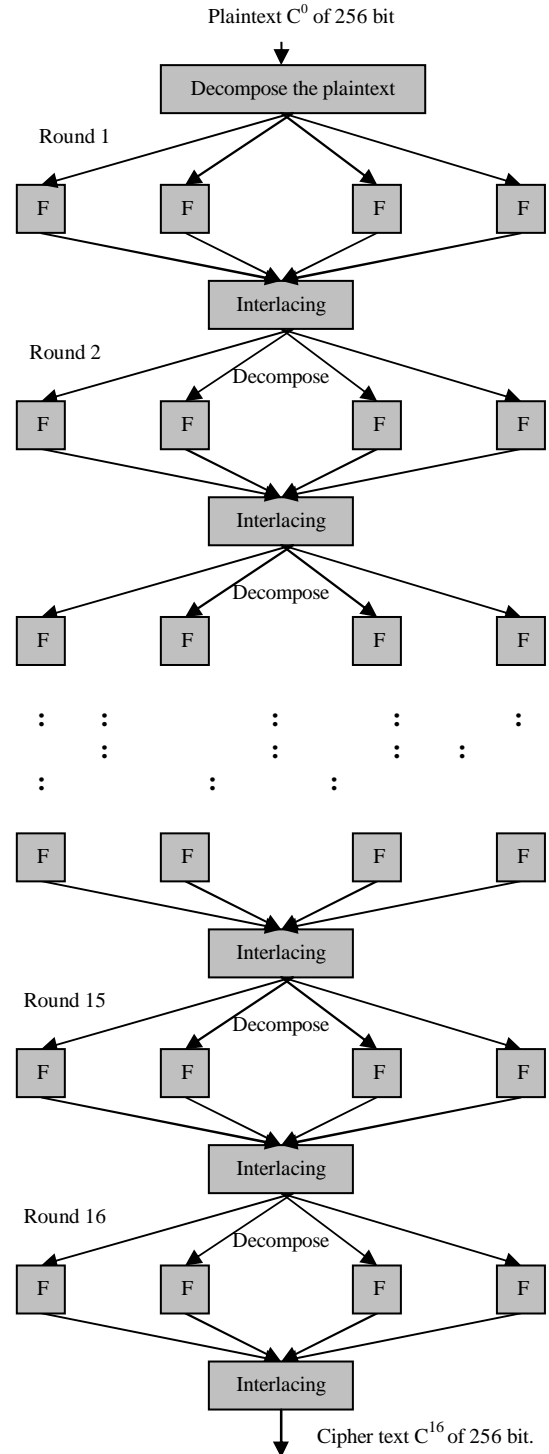
Let the key 'K' contain 16 integers, then the 8 bit binary representation of these integers yields us a block containing 128 bits. Let this block be denoted as 'k'.

Let the first 32 bits of 'k' be treated as k_1 .

The next 32 bits of 'k' be treated as k_2 . Similarly, we get two more keys ' k_3 ' and ' k_4 '.

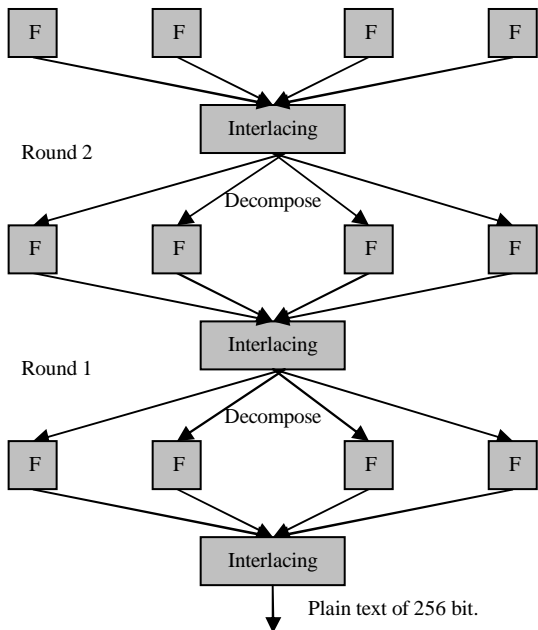
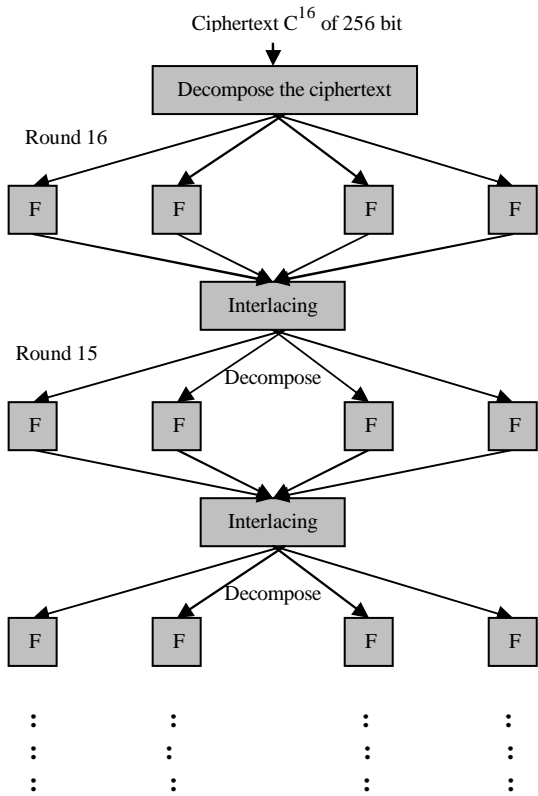
As we use four different blocks B_1, B_2, B_3, B_4 of 64 bit each for encryption, by using required transformations on k_1, k_2, k_3 and k_4 published in our previous paper, see reference [6].

The following is the process proposed for using interlacing and decomposition during encryption/decryption in feistel structure.



Encryption involving interlacing and decomposition

Note: permutations, substitutions and key generation during encryption and reverse permutations and substitutions and key generations during decryption are discussed in our paper published earlier. See reference [6].



Decryption involving interlacing and decomposition

We generate the keys for respective rounds denoted as $kr^m_1, kr^m_2, kr^m_3, kr^m_4$. Such that if kr^m_i is the round key, then 'i' indicates the block and 'm' indicates the round.

The initial plaintext of 256 bits is represented as C^0 . Decompose C^0 into four blocks of 64 bits each. This can be represented as B^0_1, B^0_2, B^0_3 , and B^0_4 . Therefore,

$B^m_i = \langle C^m \rangle$ where, 'm' indicates the round after which decomposition is performed, 'i' indicates the block number; $i = 1$ to 4 and

$\langle C^m \rangle$ indicates decomposition.

In the first round, encryption is done in the following way. We perform the required transformations on k_1, k_2, k_3 , and k_4 to get $kr^n_1, kr^n_2, kr^n_3, kr^n_4$.

$C^n_i = F_{kr^n_i} (B^m_i)$; $i = 1$ to 4 indicates i^{th} block.

'F' indicates encryption and kr^n_i indicates the round key for 'nth' round on i^{th} block and $n = m+1$.

After encryption in n^{th} round, we get ciphertext as four blocks $C^n_1, C^n_2, C^n_3, C^n_4$.

Next we perform interlacing after encryption.

$C^n = \rangle C^n_i \langle$;

Here $i = 1$ to 4, indicates the cipher block.

$n = 1$ to 16. indicates the round after which interlacing is performed.

$\rangle C^n_i \langle$, represents interlacing.

Similarly, during decryption, we proceed in the same way as discussed above, performing reverse transformations on key. See reference [6] for reverse transformations used.

4. Algorithms

4.1 Algorithm for Encryption

BEGIN

$C^0 = P$ // initialize 256 bits plaintext

for $i = 1$ to 16

{
for $j = 1$ to 4
{

$B^{i-1}_j = \langle C^{i-1} \rangle$ // Decompose

}
for $j = 1$ to 4
{

$C^i_j = F_{kr^i_j} (B^{i-1}_j)$ // Encryption

}
for $j = 1$ to 4

{
 $C^i = \rangle C^i_j \langle$ // Interlace
}

}
END

4.2 Algorithm for Decryption

BEGIN

```

C16 = cipher text // initialize 256 bits cipher text
for i = 16 to 1
{
  for j = 1 to 4
  {
    Bji = < Ci > // Decompose
  }
  for j = 1 to 4
  {
    Ci-1j = Fkrji ( Bji ) // Encryption
  }
  for j = 1 to 4
  {
    Ci-1 = > Ci-1j < // Interlace
  }
}

```

END

4.3 Algorithm for Decomposition

BEGIN

```

< Ci-1 > // during ith round
{
  j=1
  for n = 1 to 256
  {
    if ( n <= 64 )
    {
      Bi-1j [n] = Ci-1[n]
      j = j + 1
    }
    else if ( ( 64 > n ) and ( n <= 128 ) )
    {
      Bi-1j [n] = Ci-1[n]
      j = j + 1
    }
    else if ( ( 128 > n ) and ( n <= 192 ) )
    {
      Bi-1j [n] = Ci-1[n]
      j = j + 1
    }
    else if ( ( 192 > n ) and ( n <= 256 ) )
    {
      Bi-1j [n] = Ci-1[n]
      j = j + 1
    }
  }
}

```

END

4.4 Algorithm for Interlacing

BEGIN

```

> Ci-1j <
{
  for n = 1 to 64
  {
    Ci-1 [(j-1)*64 + n] = Ci-1j [ n ]
  }
}
END

```

5. Illustration Of Cipher

Consider the plaintext P = { O Lord, Please save me from evil }. Let the key K = { 155, 23, 59, 3, 111, 26, 91, 36, 77, 148, 87, 59, 118, 2, 65, 181 }.

Now the 8 bit binary representation of plaintext P and key K is as follows.

Initial plaintext C⁰ = P.

```

01001111001000000100110001101111011100100110
01000010110000100000 01010000011011000110010
10110000101110011011001010010000001110011011
00001011101100110010100100000011011010110010
10010000011001100111001001101111011011010010
000001100101011101100110100101101100 (5.1)

```

Initial key k is

```

10011011000101110011101100000011011011110001
10100101101100100100010011011001010001010111
0011101101110110000000100100000110110101(5.2)

```

Let the plaintext be decomposed into B⁰₁, B⁰₂, B⁰₃, B⁰₄. Then the respective 64 bit blocks after decomposition are as follows.

```

01001111001000000100110001101111011100100110
01000010110000100000. (5.3)

```

```

01010000011011000110010101100001011100110110
01010010000001110011. (5.4)

```

```

01100001011101100110010100100000011011010110
01010010000001100110. (5.5)

```

```

01110010011011110110110100100000011001010111
01100110100101101100. (5.6)

```

Permute the bits in key 'k' by using the random key based permutations published in our previous paper. See reference [6].

Let this permuted key be divided into four equal size blocks and used as round keys kr¹₁, kr¹₂, kr¹₃, kr¹₄. for blocks B⁰₁, B⁰₂, B⁰₃, B⁰₄. respectively.

Now we encrypt these four blocks with their respective round keys and with the help of round function 'F' as described in our previous paper published. See reference [6]. The corresponding cipher blocks $C^1_1, C^1_2, C^1_3, C^1_4$ obtained after encryption in first round are as follows.

01100001011101100110010100100000011011010110
01010010000001100110. (5.7)

0111001001101110110110100100000011001010111
01100110100101101100. (5.8)

011010000110010000000001010010010101111111
000110111110111001. (5.9)

0010001001111000111101011001000001001111110
10011010100100100010. (5.10)

Next, we need to interlace these four blocks and get a block cipher C^1 .

So that, enough confusion and nonlinearity is induced by mixing the bits of these small block ciphers.

After applying interlacing, we get the following block cipher as C^1 .

0000111011110100001000000101100000011111111
10010101111011000100000111011101000101011100
000111100001001111000000011000000000100000
111011010010100011110011111001111111110110
00011100010010100011010011110010011100100010
01110000111011100100110110010010010. (5.11)

Similarly, by using the respective round and sub keys, we continue the process up to 16 rounds and we get the following cipher.

1001111110011001000010110011010110000010111
01011000100011110111001000111110111101000101
00010001001110000001001000100110110000001001
01110100001000101100101010001111001001111100
1111011100000100101000000101001101011011000
011111000010000011000110011011101110. (5.12)

Since the process of decryption is same as the process of encryption, we get the plaintext by following the similar steps as illustrated above but with reverse permuted keys.

6. Cryptanalysis

Now, let us examine the brute force attack and the known plaintext attack on our cipher to assess the strength of the cipher. First, we show that the brute force attack is formidable and the known plaintext attack leads to a system of equations from which the unknown key cannot be determined.

6.1 Brute Force Attack

We are using 128 bit key k in each round, we divide k into four blocks, perform required transformations and get the round sub keys $kr^1_1, kr^1_2, kr^1_3, kr^1_4$ for plaintext blocks $B^0_1, B^0_2, B^0_3, B^0_4$ respectively.

According to Brute force attack, if a round key has to be guessed. We need an exhaustive search of key space

$$2^{128} \approx (2^{10})^{13} \approx (10^3)^{13} \approx 10^{39}. \quad (6.1.1)$$

Since it takes many years to test each and every key possible within such huge key space, we say that brute force attack is not possible on our algorithm as we cannot afford so many years in searching the exact key.

6.2 Known plaintext Attack

In this case, we have as many plain text – cipher text pairs as we require. In our present paper, it is worth noticing the interlacing and decomposition concepts introduced which handle the known plaintext attack. Let us first understand how classical feistel cipher is prone to known plaintext attack and then will discuss how our modified feistel cipher tackles this problem.

According to classical feistel cipher network, the problem is with a particular set of bits, which always undergo into similar transformations in every successive round. For example, the first six bits always go into the first substitution box. Therefore, if we have enough plaintext cipher text pairs, one can easily guess the values used in a substitution box ignoring the other substitution boxes. Similarly, one will be able to guess the key bits also. This problem does not exist in our modified algorithm because; we are using four independent blocks of encryption in each round. It is ensured that bits after a particular round will not enter into the same substitution boxes, will not use the same permutations and key. This is due to interlacing and decomposition concepts, which allow the scattering of bits into four different blocks. Thus, interlacing and decomposition allow us to mix the bits properly and it helps us in introducing high nonlinearity in the algorithm.

6.3 Avalanche Effect

Let the plaintext be "O Lord, Please save me from evil". By following the process of encryption, we get the cipher.

1001111110011001000010110011010110000010111
01011000100011110111001000111110111101000101
00010001001110000001001000100110110000001001
01110100001000101100101010001111001001111100
1111011100000100101000000101001101011011000
011111000010000011000110011011101110. (6.3.1)

Now let the plaintext be fixed, but change the key by one bit. This can be done by changing the number "155" to "156" in key 'K', since 155 and 156 differ by one bit. Now

by using this new key 'k' we encrypt the same plaintext and we obtain the corresponding cipher as

```
0011001001010101101011100111001011111110110
01010011101110000101001000100010100001011101
01010101111010111100000111000001010001111010
00110101011110010110101101101010010101101010
01010101110010001011011100111100011001000100
100010011001010011010010101111010011 (6.3.2)
```

Comparing (6.3.1) and (6.3.2), we notice that the two cipher blocks differ by 125 bits out of the total 256 bits. This shows that the algorithm exhibits strong avalanche effect.

In the second case, let the key 'K' be fixed, But change the plaintext. So that, the new plaintext and the original one differ by exactly one bit. This can be accomplished by changing the first character of the plaintext from 'O' to 'P', because, ASCII values of 'O' and 'P' differ by one. We get the cipher text from this new plaintext as

```
11011100010001000100000100011000001000000101
00100100001110111010101111000001101100100110
11110010110010010000111001111001000111101000
00010001010011100100100000111000101001000101
00101010011111010011010110100010010010001100
001101101011001011100010001010101010 (6.3.3)
```

On comparing (6.3.1) and (6.3.3), we notice that the two cipher blocks differ by 125 bits out of 256 bits. This shows, that the interlacing and decomposition introduced in our encryption algorithm exhibits good avalanche affect.

7. Computational Results and Conclusion

In this paper, we have developed a block cipher of 256 bits. The plaintext is of 32 characters and each character is represented with its 8 bits binary equivalent. The key contains 16 integers which converted into its 8 bits binary equivalent. The algorithms used for encryption, decryption, decomposition, interlacing etc. are all written using C language.

From the cryptanalysis presented, we found that, brute force attack is not possible. There is enough confusion and diffusion introduced in the encryption algorithm through the concepts of interlacing and decomposition. This is proved by the avalanche effect that is shown in (6.3). By using interlacing and decomposition, a 256 bit block, is broken

Acknowledgement

The authors are very thankful to Prof. Depanwita Roy Chaudhury, IIT Kharagpur, India, for giving necessary suggestions and for her valuable inputs given while writing this paper. The authors are very thankful to the management of Sreenidhi Institute Of Science and Technology, for their support and encouragement given during this research work.

into 4 equal parts of 64 bit blocks so that, cipher bits obtained after each round scatter into different blocks in the next round. By doing so, the cryptanalysis part becomes more difficult as the final cipher text obtained will depend on different substitution boxes and different transformations

References

- [1] William Stallings, "Cryptography and Network Security: Principles & Practices", Third edition, 2003, Chapter 2 and 3.
- [2] Feistel. H. "Cryptography and Computer Privacy", Scientific American, Vol. 228, No. 5. pp 15 – 23, 1973.
- [3] Feistel, H., Notz W. and Smith. J. "Some cryptographic Techniques for machine to machine data communications", Proceedings of the IEEE, Vol. 63, No. 11, pp 1545 – 1554, Nov 1975.
- [4] "Avalanche Characteristics of Substitutions – permutation Encryption Networks" Tavares S. Heys H. IEEE Transactions on Computers 44 (9): 1131 – 1139, 1995.
- [5] Shakir M. Hussain and Naim M. Ajilouni, "Key based random permutation", "Journal of Computer Science 2(5): 419 – 421, 2006. ISSN 1549 -3636.
- [6] K. Anup Kumar and S. Udaya Kumar, "Block cipher using key based random permutations and key based random substitutions", "International Journal of Computer Science and Network Security", Seoul, South Korea. ISSN: 738-7906. Vol. 08, No. 3, March 2008. pp. 267-277.

Authors Profile

K. Anup Kumar is working as an Associate Professor in the Department Computer Science and Engineering, Sreenidhi Institute of Science and Technology. He is pursuing his PhD in the area of information security, Under the guidance of Prof. V.U.K. Sastry from Jawaharlal Nehru Technological University, Hyderabad, India. He published two papers in international Journals. He is interested in the research areas like: cryptography, Steganography, and Parallel processing systems.

Prof. V.U.K. Sastry is working as the Director school of computer science and informatics and as Dean R & D CSE Department in Sreenidhi Institute of Science and technology. Hyderabad, India. He has successfully guided many PhD's and his research interests are: information security, Image processing and Data warehousing - data mining. He is the reviewer of many international journals.