ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

An Enhanced AES Algorithm Based on Variable Sbox And 200 Bit Data Block

Shivedra Shekhar[1], Pushkar Singh[2], Manjit Jaiswal[3]

Student, Dept. of C.S.E., Guru Ghasidas VishwaVidyalaya, Bilaspur, Chhattisgarh, India[1,2]

Asst. Professor, Dept. of C.S.E., Guru Ghasidas Vishwavidyalaya, Bilaspur, Chhattisgarh, India[3]

ABSTRACT: This paper suggests a new Advance Encryption Standard (AES) [3][9] encryption technique. New algorithm uses a 200 bit size[2] block encryption scheme and a key dependent substitution box which varies according to the 200 bit key provided by the user. Algorithm uses 200 bit key in the form of a 5*5 matrix[2] instead of using general AES 128 bit, 192 bit and 256 bit size encryption technique[3][8]. The new suggested varying substitution box [6] provides a better and secure way to the encryption of data. Substitution box is being varied using s-box rotation based on the key used for encryption..

KEYWORDS: AES, Substitution box, Encryption techniques, Block encryption, S-box rotation

I. INTRODUCTION

Now a day's technology has evolved very much, so these days securing someone's private data is very important, to do so we need encryption or ciphering of the data they have, to make data secure from illegal copying, distribution and attackers. Ciphering is very important operation to preserve the confidentiality of digital images [5] transmitted over public network especially with rapidly growth in communication techniques. One of the strongest encryption techniques is AES [7]. This technique is used by government agencies and banks to secure their data. This technique is only vulnerable to brute force attack which is almost impossible to be used in breaking as it requires too much time in checking all the possibilities of keys that can be generated by given constraints.

The Advance Encryption Standard (AES) [9][10] algorithm was proposed by Joan Daemon and Vincent Rijmen named as "Rijndael". They submitted their algorithm in response to the call of National Institute of Standards and Technology (NIST) to replace the obsolete Data Encryption Standard (DES)[4][8] in the year 1997. Algorithm was designed for the use of 128 bit, 192 bit and 256 bit length symmetric key [1] with limited block length of 128 bits for encryption.

AES, the Advanced Encryption Standard, is a substitution-cum-permutation block cipher based on the forward S-Box built for encryption and the inverse S-Box built for decryption. In AES encryption, to introduce non-linearity [9], the 8-bit S-Box was generated using modular arithmetic. 128 bit block of data is given to the AES as input for both encryption and decryption. The key is provided as a 4*4 matrix. Total rounds of AES algorithm varies according to the key size used. Rounds are 10, 12 and14 for 128 bit, 192 bit and 256 bit key size respectively. Rijndael rounds contain four different stages namely SubByte Transformation, ShiftRow, MixColumn, and AddRoundKey [4] [5].

1. SubByte Transformation: This stage substitutes each cell of 4*4 matrix of 128 bit block with a Substitution box (Sbox) cell through some predefined operations. This provides non linearity and confusion and also known as Sbox substitution stage.
2. ShiftRow: This stage rotates each row of the block according to its row index providing inter column diffusion. This is also known as rotation of rows.
3. MixColumn: Predefined operations with another fixed 4*4 polynomial matrix are performed in this stage and value of each cell is changed according to the operation. Here bytes are treated as polynomials rather than numbers. This provides inter–byte diffusion and known as linear combination stage.
4. AddRoundKey: XOR operation is performed between the data block and the round key which is different for each round. This also provides confusion.

AES algorithm also incorporates a key generating phase which generates keys for each round of the AES separately. This key generation phase uses keys of previous rounds to create the keys for next round using operations like RotateWord, SubByte and XORing with “RoundConstant” matrix (which is already defined in AES).

For the decryption of data we just have to reverse all of the above stages with encrypted block as input. We also have to use inverse sbox in place of sbox in Sbox substitution stage.

In the last round of both encryption and decryption, algorithm does not have MixColumn stage.

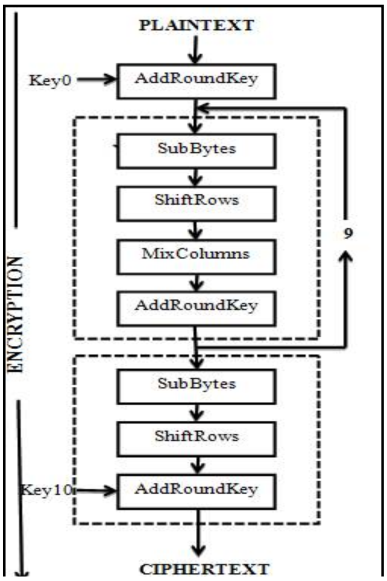


Figure 1. AES Encryption

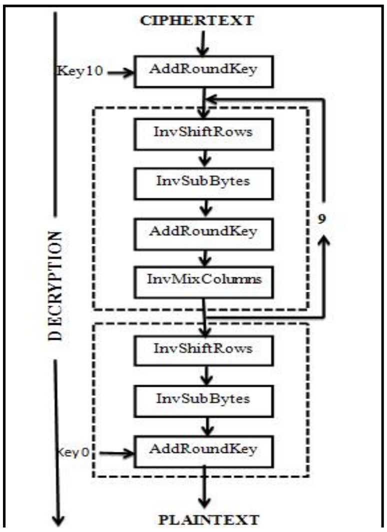


Figure 2. AES Decryption

II. RELATED WORK

Now a days, secure message transmission has become an essential task. So a lot of researches are going on, in the field of encryption and other field of cryptography. A lot of research papers have been published in various journals like IEEE etc.

We have selected AES algorithm for encryption for a reason that is it is fast and more secure in comparison to DES, 3DES encryption algorithm. Now days due to invention of fast systems the bruit force attack has become more effective. But due to enhanced range of keys these effects are still not feasible for breaking the security of AES algorithm [3].

AES is a block cipher symmetric encryption algorithm that is widely used for encryption. In AES a particular block of the plain text is passed through the series of steps that are shown in the given

Some issues have already been resolved through these research papers but better solutions of these issues are still required. In a paper published in science direct, The AES was parallelized using the concepts of neural network.

A new algorithm containing different key size and enhanced S-Box has also been proposed in order to enhance the security and enhance the speed of the algorithm [6]. Some work related to space complexity is also going on and is major field of research.

III. PROPOSED ALGORITHM

Rijndael's algorithm uses basic 128 bit block of data as input and a fixed and predefined substitution box for its operations, The new algorithm suggests the use of 200 bit block size of data and same size of key in 5*5 matrix format and a Substitution box which varies according to the key input given by the user which will make this new algorithm more robust and secure. The proposed algorithm contains one extra stage for calculation of sbox based on the 200 bit key [2].

1. **Variable Substitution box**: The original AES algorithm uses static 16*16 sbox for encryption but our proposed algorithm will use a varying 16*16 sbox based on the cipher key and will be changed in each time in different encryptions automatically [6].

The sbox used in encryption is calculated with the help of original Rijndael's fixed sbox and 200 bit key given by the user.

- Procedure for sbox generation: To generate a new Sbox which will be used throughout the encryption process, First of all we have to take XOR of elements of each cell of 5*5 cipher key provided by the user. Then the elements of original sbox are rotated by the value "result" times.

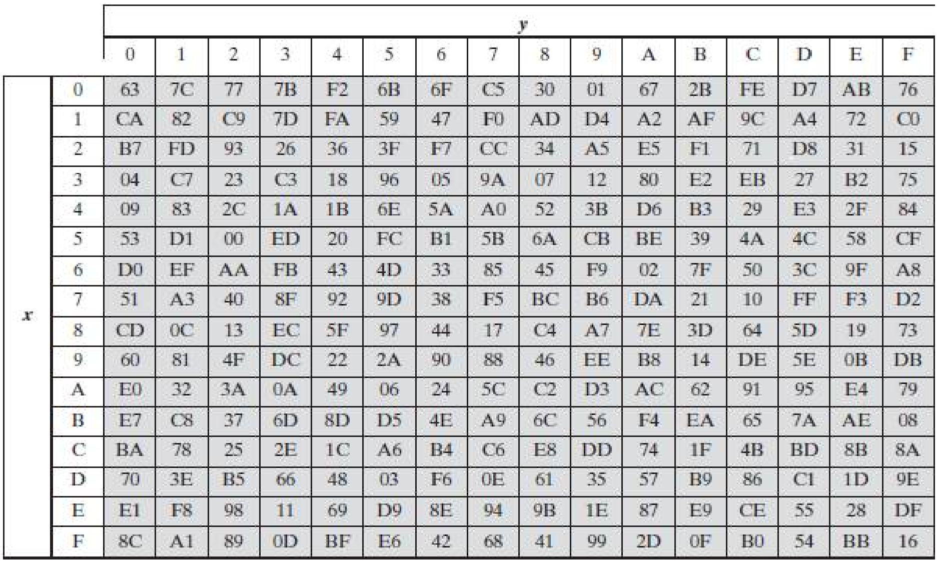
  e.g. cipher key is: 7D558EAC0E403CD82D95275 E371992420FF0FF700700A00A00 (in Hex)

  Apply XOR operation on all the bytes.

  9F (HEX) = 7D^55^8E^AC^0E^40^3C^D8^2D^95^27^5E^37^19^92^42^0F^F0^FF^70^07^00^

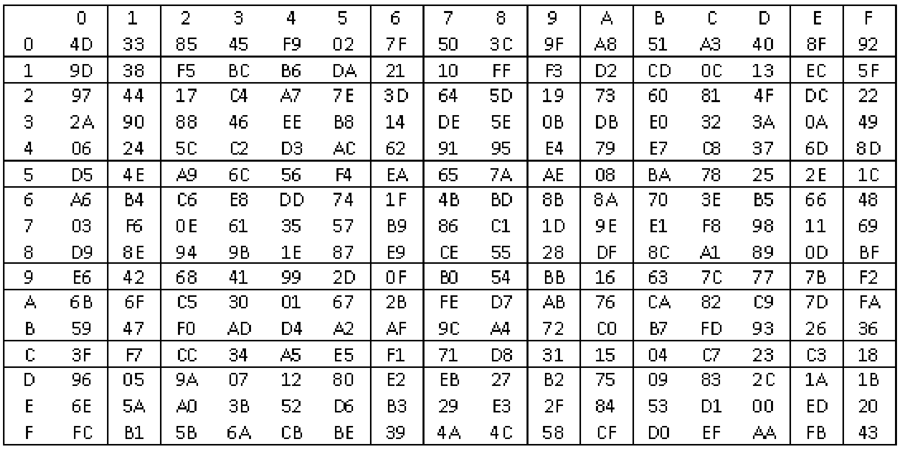
  A0^0A^00

  So the original sbox is rotated by 159(decimal equivalent of 9F) times. And this rotated sbox will be used in the encryption.



		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Fig 3. Rijndael's sbox.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4D	33	85	45	F9	02	7F	50	3C	9F	A8	51	A3	40	8F	92
1	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2	CD	0C	13	EC	5F
2	97	44	17	C4	A7	7E	3D	64	5D	19	73	60	81	4F	DC	22
3	2A	90	88	46	EE	B8	14	DE	5E	0B	DB	E0	32	3A	0A	49
4	06	24	5C	C2	D3	AC	62	91	95	E4	79	E7	C8	37	6D	8D
5	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08	BA	78	25	2E	1C
6	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A	70	3E	B5	66	48
7	03	F6	0E	61	35	57	B9	86	C1	1D	9E	E1	F8	98	11	69
8	D9	8E	94	9B	1E	87	E9	CE	55	28	DF	8C	A1	89	0D	BF
9	E6	42	68	41	99	2D	0F	B0	54	BB	16	63	7C	77	7B	F2
A	6B	6F	C5	30	01	67	2B	FE	D7	AB	76	CA	82	C9	7D	FA
B	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0	B7	FD	93	26	36
C	3F	F7	CC	34	A5	E5	F1	71	D8	31	15	04	C7	23	C3	18
D	96	05	9A	07	12	80	E2	EB	27	B2	75	09	83	2C	1A	1B
E	6E	5A	A0	3B	52	D6	B3	29	E3	2F	84	53	D1	00	ED	20
F	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF	D0	EF	AA	FB	43

Fig 4. New Sbox after rotation.

Based on the above procedure a new sbox is created each time when a new cipher key is given for the both encryption and decryption process.

2. **Use of 200 bit key and block:** Our proposed algorithm uses a 200 bit block instead of original 128 bit block for input data and it also uses a 200 bit size cipher key in form of 5*5 matrix which brings changes in stages of each round.

- SubByte Transformation: Due to use of 200 bit key there will be no change in the sbox substitution stage as the size of new substitution box remains same.
- ShiftRow: In this stage instead of 4 row shift operations (as in Rijndael algorithm), 5 row shift operation occurs.

DOI: 10.15680/IJIRCCE.2016. 0404026

Before ShiftRow operation:

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$	$S_{0,4}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	$S_{2,4}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$	$S_{3,4}$
$S_{4,0}$	$S_{4,1}$	$S_{4,2}$	$S_{4,3}$	$S_{4,4}$

After ShiftRow operation:

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$	$S_{0,4}$
$S_{1,4}$	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,3}$	$S_{2,4}$	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$
$S_{3,2}$	$S_{3,3}$	$S_{3,4}$	$S_{3,0}$	$S_{3,1}$
$S_{4,1}$	$S_{4,2}$	$S_{4,3}$	$S_{4,4}$	$S_{4,0}$

- MixColumn: The polynomial matrix is changed from 4*4 matrix to 5*5 and values of cells are changed accordingly. The polynomial used for multiplication is as following:-

  $2x^4+4x^3+3x^2+x+1$

So the polynomial matrix corresponding to the above polynomial used in mixed column step is:-

02	04	03	01	01
01	02	04	03	01
01	01	02	04	03
03	01	01	02	04
04	03	01	01	02

And the corresponding inverse polynomial matrix used for decryption is:-

E0	7D	09	8A	4C
4C	E0	7D	09	8A
8A	4C	E0	7D	09
09	8A	4C	E0	7D
7D	09	8A	4C	E0

The polynomial matrix is multiplied with 200 bit plain text and the resultant block is fed to the next step(add rounded key).

AddRoundKey Transformation: In AddRoundKey transformation, a rounded key is added to the State by bitwise Exclusive-OR (XOR) operation.

IV. PSEUDO CODE

A. S-BOX CALCULATIONSIMULATION RESULTS

```
Step1. Partition the 200 bit key into 25 partitions of 8 bits each.
Step2. Xor these sections together to calculate a number (let ‘no’).
Step3. Shift the original S-box by ‘no’ number of times towards right.
```

B. SubByte operation

```
Step 1: loop until the entire block is replaced:
Step 2: take each cell (1 Byte) for replacement. And set the row no by the first nibble of the cell value and the col no. by the second nibble of the cell value.
Step3: Replace the considered cell by the value at ‘row no.’ row and ‘col no.’ column in the modified Sbox calculated in Sbox calculation step.
Step 4: if the entire 200 bit is replaced then exit else go to step2.
```

C. ShiftRow Operation

```
Step 1: set I=0 and take the first row for consideration.
Step 2: Rotate the considered row by I number of times towards right.
Step 3: set I =I+1
Step 4: if I< 5 then go to Step 2.
Step 5: Exit.
```

D. MixColumn Operation

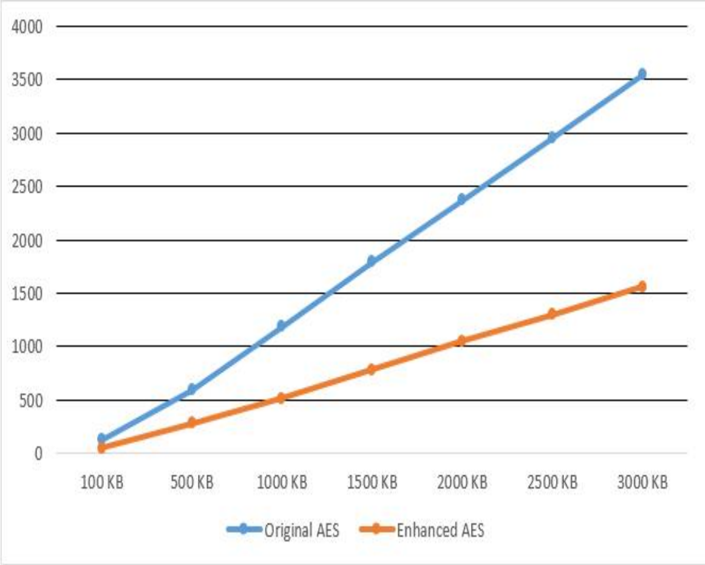
```
Step 1: Set I=0
Step 2: Set J=0
Step 3: Perform multiplication operation between $^{th}$row of data block (p_t[5][5]) and $J^{th}$ column of MixColumn matrix.
Step 4: Set J=J+1
Step 5: if J< 5 then go to step 3.
Step 6: Set I = I+1
Step 7: if I<5 then go to step 2.
Step 8: exit.
```

E. AddRoundKey Operation

```
Step 1: Set I=0
Step 2: Set J=0
Step 3: p_t[i][j]=p_t[i][j] Xor key[i][j]       //p_t[5][5] is data matrix and  key[5][5] is key matrix.
Step 4: Set J=J+1
Step 5 if j<5 then go to step 3.
Step 6: I=I+1
Step 7: if i<5 then go to step 2.
Step 8: exit.
```

IV. SIMULATION RESULTS

We have compared our proposed algorithm and original algorithm on different set of block of data for encryption time and our proposed algorithm presents good results with respect to the running or encryption time of encrypting block of data, which is shown in the graph.



Our algorithm uses 200 bit block of data for encryption at a time which allows it to encrypt more data than the normal algorithm in a single round making it faster than the original algorithm. Graph shows that the enhanced algorithm takes 50% time of the original algorithm for large data.

New algorithm uses a variable substitution box in place of a static one which hopefully will provide a better and more secure encryption of the block of data.

V. CONCLUSION AND FUTURE WORK

In our proposed work we have improved the efficiency of AES algorithm by variable s-box, 200 bit data block and key. We have also encrypted the images in jpeg format into the text format. So finally we conclude that the new improvised algorithm works well and saves a lot of encryption time and provides more security to the encrypted document.
In our future work we will try to implement AES algorithm on multi-threaded system.
We will also implement AES algorithm on GPU (Graphics Processor Unit) for parallelisation purpose which will make AES algorithm more fast and reliable, as many tasks will be carried out parallel to reduce the time of encryption. This will allow us to encrypt more data in less time.

REFERENCES

[1] Anita Ganpati, Narender Tyagi "Comparative Analysis of Symmetric Key Encryption Algorithms" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 8, August 2014, ISSN: 2277 128X.

[2] Ritu Pahal, Vikas kumar "Efficient Implementation of AES" International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 7, July 2013, ISSN: 2277 128X

[3] Supriya, Gurpreet Singh "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) forInformation Security" International Journal of Computer Applications (0975 – 8887) Volume 67–No.19, April 2013.

[4] Dr. Prerna Mahajan & Abhishek Sachdeva "A Study of Encryption Algorithms AES, DES and RSA for Security" Global Journal of Computer Science and Technology Network, Web & Security, Volume 13 Issue 15 Version 1.0 Year 2013, ISSN: 0975-4172.

[5] Salim M. Wadia, Nasharuddin Zainala "Rapid Encryption Method Based on AES Algorithm for Grey Scale HD Image Encryption" The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013).

[6] Julia Juremi, Ramlan Mahmod, Salasiah Sulaiman, Jazrin Ramli "Enhancing Advanced Encryption Standard S-Box Generation Based on Round Key" International Journal of Cyber-Security and Digital Forensics (IJCSDF) 1(3): 183-188 the Society of Digital Information and Wireless Communications (SDIWC) 2012 (ISSN: 2305-0012).

[7] Nidhi Singhal, J.P.S.Raina "Comparative Analysis of AES and RC4 Algorithms for Better Utilization" International Journal of Computer Trends and Technology- July to Aug Issue 2011, ISSN: 2231-280.

[8] Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani, "New Comparative Study between DES, 3DES and AES within nine factors" JOURNAL OF COMPUTING, VOLUME 2, ISSUE 3, MARCH 2010, ISSN 2151-9617.

[9] Stallings W, Cryptography and Network Security, Pearson Prentice Hall, New Delhi, 6th Impression, (2008).

[10] Foruzan BA, Cryptography and Network Security, Tata McGraw-Hill, New Delhi, Spl. Indian Edition, (2007).

BIOGRAPHY

Mr. Manjit Jaiswal is currently working as a Assistant professor in the Computer Science and engineering Department, Institute of technology, Guru Ghasidas Vishvavidyalaya A Central University Bilaspur,Chhattisgarh,India. He received Master of Technology (M-Tech) degree in 2012 from MANIT, Bhopal, M.P., India. His research interest fields are Parallel computing,Firewall,Network Security,Distributed Sstem, Algorithms,Routing Algorithm etc.

Mr. Pushkar Singhis Pursuing his Bachelor of Technology (B-Tech) in Computer Science and engineeringDepartment, Institute of technology, Guru Ghasidas Vishvavidyalaya,A Central University, Bilaspur, C.G., India.His research interests are Network Security, Algorithms etc.

Mr. Shivendra Shekharis Pursuing his Bachelor of Technology (B-Tech) in Computer Science and engineering Department, Institute of technology, Guru Ghasidas Vishvavidyalaya, Bilaspur, C.G., India.His research interests are Network Security, Algorithms etc.