International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5, June 2019

Modified Version of Playfair Technique to Enhance the Security of Plaintext and Key Using Rectangular and Substitution Matrix

Kamal Kumar Gola, Manish Dhingra, Rahul Rathore

Abstract*: To make the data more secure there are several techniques. Most of the techniques have modified many times. Most authors used those modified techniques in their algorithm. Playfair technique is one of them. Playfair technique has also modified several time. Most authors used 6*6 matrix which include alphabets in lowercase, integer value as well as special symbols, most 7*7 matrix which include alphabets in uppercase and in lower case both, integer value as well as special symbols. We used 9*9 matrix which includes alphabets in lowercase or in uppercase both, integer value as well as list of operators and putting in the matrix according to their value in ASCII code. ASCII code stands for American Standard Code for Information Interchange. This work uses ASCII code to provide more security. For dual security, a substitution matrix is also used so that no one can understand it easily, even decoder have to do more and more efforts for decrypting the cipher text. Hence with the help of this technique we make our data safe.

***Index Terms*: Rectangular Matrix, Substitution Matrix, Playfair, Encryption and Decryption.**

I. INTRODUCTION

Cryptography is the science of text modification becoming unreadable. Cryptography has patterns like in steganography and image processing [1] [2]. Data security is required because of the large number of cybercrimes [3].It is called the encryption technique where plaintext is randomized using a key to be ciphertext [4] [5]. Compression is also a cryptographic model that compresses the message content [6] [7].If someone does not have a decryption key then the person cannot understand the content of the text [8]. Decryption is the process of returning ciphertext to plaintext. The probability of retrieving the original script by someone who has not had a decryption key is very small [9].Play fair cipher is widely used and quite useful in its era. Play fair cipher is a classical cryptographic algorithm that belongs to a Polygram cipher where play index is con-verted to a Polygram form and a decryption encryption process perform for the poly-graph. In [10] the authors have used 5*5 matrixes. According to this the key arrangement inside the square extends by adding the sixth and sixth rows. The sixth base is the first line while the sixth columns contain the first column. In general, the key used in a series of words that are easy to understand. The use of playfair cipher method on text encoding is good enough because the key matrix used has a small possibility to be solved. Super playfair cipher is best used for symmetric cryptographic type. The bigram substitution technique on the key matrix has a small chance to solve. Each bigram change, the matrix pattern changes to the key [10].

Manuscript published on 30 June 2019.
***** Correspondence Author (s)
Kamal Kumar Gola*, Faculty of Engineering, Teerthanker Mahaveer University, Moradabad, 244001, India.
Manish Dhingra, Faculty of Engineering, Teerthanker Mahaveer University, Moradabad, 244001, India.
Rahul Rathore, Faculty of Engineering, Teerthanker Mahaveer University, Moradabad, 244001, India.

II. RELATED WORK

In [11] the author used in his research a 6*6 matrix which consist the alphabets in uppercase as well as integer value. The author used loop process for generating key-words and generates four keywords. These keywords have used for the encryption and decryption process. In [12] the author used in his research a 6*6 matrix which consist the alphabets as well as integer value. For more security of data the author used excess 3-code and generate rectangular matrix. In this work he encrypted the key. He used Ceaser Cipher technique for the encryption of key. In [13] the author used in his research a 12*8 matrix which consist the alphabets, integer value as well as special symbol. The author divided his work into two parts. In the first part he extended the size of matrix and in the second part converted into ASCII code (American Standard Code For Information Interchange).further this he used RSA public key encryption for the encryption of key. In [14] the author used a RSA digital signature algorithm. Digital signature algorithm provided the data services. In his algorithm author encrypted the key during transmission of data. In [14] the author used in his research a 10*8 matrix which consist the alphabets (in uppercase and in lowercase both), integer value from 0 to 9, list of operators as well as list of brackets. This work has divided into two parts. In the first part the key encrypted with the help of ASCII code and in the second part the concept of digital signature used [15]. A great deal of research has been done on various characters of Playfair ciphers. In [16] the authors have implemented a 5*5 matrix. While in [17] the authors have implemented an 8*8 matrix on DNA encoded data.

III. PROPOSED WORK

A. *Generation of Rectangular Matrix*

i) First place all the alphabets of key in a given matrix of from left to right and top to bottom with no repeating alphabets, integer value and opera-tors in **Table 1.**

ii) Now fill all the remaining entries of matrix 9*9 with remaining alpha-bets, integer value and operators according to their ASCII code value column wise from **Table 2**.

iii) Convert all the entries of Table 1 into its equivalent integer value ac-cording to ASCII code.

iv) Now convert all the entries into its equivalent binary number of 7 bits.

v) Now putting the equivalent value of the binary number with the help of substitution matrix **Table 3**.

a) First read first and last bit and converted into decimal value.

b) With the help of this decimal value and the integer value which is equivalent to the binary number we get another integer value in the substitution matrix **Table 3.**

c) Now place this value in a table known as **Table 4.**

d) Now convert the values of **Table 4** according to the ASCII code which further produce the **Table 5**. This matrix will be used for encryption purpose.

B. *How Algorithm Works*

i) First divide the plain text into a pair of two. If there are repeated letter then break it and use an arbitrary letter x. For example: Hello then we divide it as: He lx lo

ii) We will check the letter which we want to be encrypted in the **Table 1.** Now take the position of the respective ciphertext from **Table 1.** With the help these position, take the final ciphertext from **Table 5.** If the letters appears on the same row in Table 1 replacing it with the immediately right in the respective table.

iii) If the letters appear on the same column of the **Table 1**, replace them with letters immediately below in table.

iv) It the letters neither in the same row nor the same column but at that column or row which form rectangle replacing it by making rectangle.

C. *Key Encryption Process*

i) First convert the each character of key MyEmailid␣ into its equivalent ASCII value.

ii) Now convert the value into its equivalent binary number of 7 bits.

iii) Convert the value into equivalent gray code.

iv) Now convert the gray code into its equivalent integer value.

v) Finally convert the values into respective ASCII code which produce the final encrypted key for transmission.

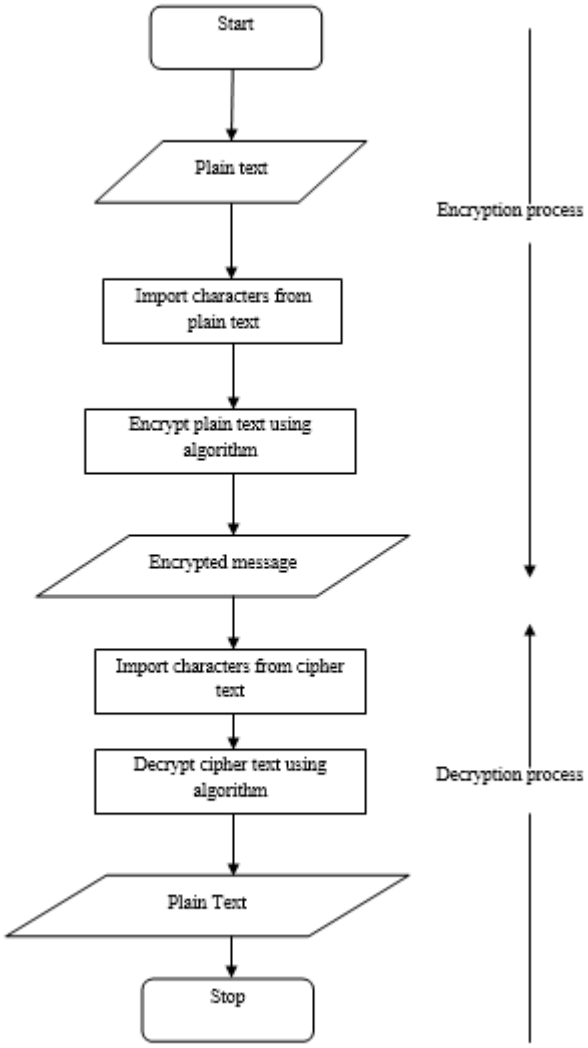

Fig 1. Flow Chart of Encryption-Decryption Process

Table 1. Base Table

M	y	E	m	A	i	l	d	␣
!	“	#	$	%	&	‘	(	)
*	+	,	_	.	/	0	1	2
3	4	5	6	7	8	9	@	A
B	C	D	F	G	H	I	J	K
L	N	O	P	Q	R	S	T	U
V	W	X	Y	Z	[	]	b	c
e	f	g	h	J	k	n	o	p
Q	r	s	t	U	v	w	x	z

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Table 2. Symbols Representation

␣	)	2	A	J	S	]	i	r
!	*	3	B	K	T	a	j	s
“	+	4	C	L	U	b	k	t
#	,	5	D	M	V	c	l	u
$	_	6	E	N	W	d	m	v
%	.	7	F	O	X	e	n	w
&	/	8	G	P	Y	f	o	x
‘	0	9	H	Q	Z	g	p	y
(	1	@	I	R	[	h	q	z

␣ Blank Space

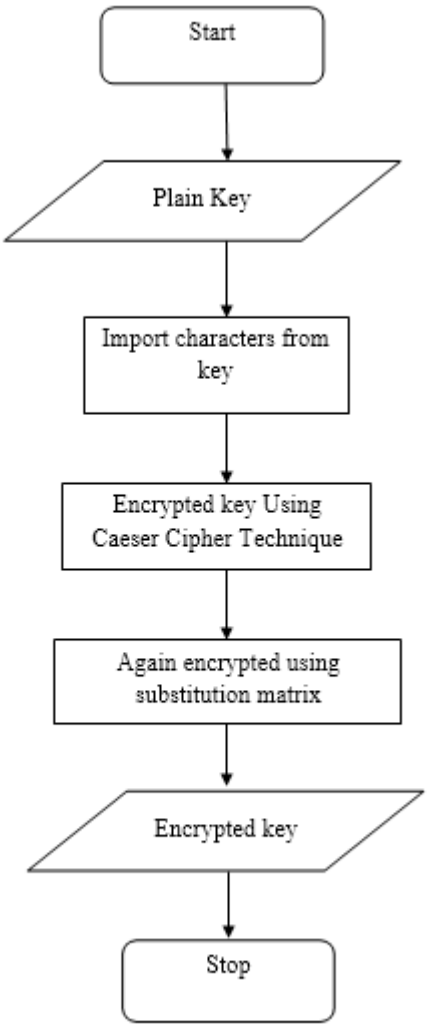

Fig 2. Flow Chart for Key Encryption Process

IV. IMPLEMENTATION

4.1 Generation of Rectangular Matrix and Encryption
Suppose we have 9*9 rectangular matrix. In a given example key is MyEmailid and plain text is kkgolaa1503@gmail.com where ␣ represents the blank space.

i) First place all the alphabets of key in a given matrix from left to right and top to bottom with no repeating alphabets, integer value and opera-tors.

ii) Now fill all the remaining entries of matrix 9*9 with remaining alpha-bets, integer value and list of operators according to the ASCII code from Table 2.

M	**y**	**E**	**m**	**A**	**i**	**l**	**d**	**␣**
!	**“**	**#**	**$**	**%**	**&**	**‘**	**(**	**)**
*****	**+**	**,**	**_**	**.**	**/**	**0**	**1**	**2**
3	**4**	**5**	**6**	**7**	**8**	**9**	**@**	**A**
B	**C**	**D**	**F**	**G**	**H**	**I**	**J**	**K**
L	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**
V	**W**	**X**	**Y**	**Z**	**[**	**]**	**b**	**c**
e	**f**	**g**	**h**	**J**	**k**	**n**	**o**	**p**
Q	**r**	**s**	**t**	**U**	**v**	**w**	**x**	**z**

iii) Now convert all the filled values according to ASCII code.

77	**121**	**69**	**109**	**97**	**105**	**108**	**100**	**32**
33	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**
42	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
51	**52**	**53**	**54**	**55**	**56**	**57**	**64**	**65**
66	**67**	**68**	**70**	**71**	**72**	**73**	**74**	**75**
76	**78**	**79**	**80**	**81**	**82**	**83**	**84**	**85**
86	**87**	**88**	**89**	**90**	**91**	**93**	**98**	**99**
101	**102**	**103**	**104**	**106**	**107**	**110**	**111**	**112**
113	**114**	**115**	**116**	**117**	**118**	**119**	**120**	**122**

iv) Now convert all the entries into its equivalent binary number.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

1001101	**1111001**	**1000101**	**1101101**	**1100001**	**1101001**	**1101100**	**1100100**	**0100000**
0100001	**0100010**	**0100011**	**0100100**	**0100101**	**0100110**	**0100111**	**0101000**	**0101001**
0101010	**0101011**	**0101100**	**0101101**	**0101110**	**0101111**	**0110000**	**0110001**	**0110010**
0110011	**0110100**	**0110101**	**0110110**	**0110111**	**0111000**	**0111001**	**1000000**	**1000001**
1000010	**1000011**	**1000100**	**1000110**	**1000111**	**1001000**	**1001001**	**1001010**	**1001011**
1001100	**1001110**	**1001111**	**1010000**	**1010001**	**1010010**	**1010011**	**1010100**	**1010101**
1010110	**1010111**	**1011000**	**1011001**	**1011010**	**1011011**	**1011101**	**1100010**	**1100011**
1100101	**1100110**	**1100111**	**1101000**	**1101010**	**1101011**	**1101110**	**1101111**	**1110000**
1110001	**1110010**	**1110011**	**1110100**	**1110101**	**1110110**	**1110111**	**1111000**	**1111010**

Table 3. Substitution Table

		32	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**
Outer Bits	**0**	120	122	118	116	114	112	110	108	106	104	102	100	98	90	88	86	84	82	80	78
	1	33	35	37	39	41	43	45	47	49	51	53	55	57	65	67	69	71	73	75	74
	2	117	119	122	115	113	111	109	107	105	103	101	99	97	93	91	89	87	85	83	81
	3	100	98	90	88	86	84	82	80	78	76	74	72	70	68	66	64	56	54	52	50
		52	**53**	**54**	**55**	**56**	**57**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**	**76**	**77**
Outer Bits	**0**	76	74	72	70	68	66	56	64	54	52	50	48	46	44	42	40	38	36	34	32
	1	72	70	68	66	64	56	54	52	50	48	46	44	42	40	38	36	34	32	77	79
	2	79	77	75	73	71	69	67	65	57	55	53	51	49	47	45	43	41	39	37	35
	3	48	46	44	42	40	38	36	34	32	102	104	106	108	110	112	114	116	118	120	122
		78	**79**	**80**	**81**	**82**	**83**	**84**	**85**	**86**	**87**	**88**	**89**	**90**	**91**	**93**	**97**	**98**	**99**	**100**	**101**
Outer Bits	**0**	35	33	37	39	41	43	45	47	49	51	53	55	57	65	67	69	71	73	75	77
	1	81	83	85	87	89	91	93	97	99	101	103	105	107	109	111	113	115	117	119	121
	2	34	32	33	36	38	40	42	44	46	48	50	52	54	56	64	66	68	70	72	74
	3	99	97	93	91	89	87	85	83	81	79	77	75	73	71	69	67	65	57	55	53
		102	**103**	**104**	**105**	**106**	**107**	**108**	**109**	**110**	**111**	**112**	**113**	**114**	**115**	**116**	**117**	**118**	**119**	**120**	**121**	**122**
Outer Bits	**0**	79	81	83	85	87	89	91	93	97	101	99	105	103	107	111	109	113	115	117	119	121
	1	122	120	118	116	114	112	110	108	106	104	102	100	98	90	88	86	84	82	80	78	76
	2	76	78	80	82	84	86	88	90	98	102	100	104	106	108	110	112	114	116	118	121	120
	3	51	49	47	45	43	41	39	37	35	33	101	103	107	105	109	111	121	117	115	113	119

v) To find the Table 4, puts the values by using substitution matrix according to the rule as discussed above.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Table 4. Values after applying substitution operation.

122	**113**	**106**	**37**	**67**	**45**	**88**	**72**	**120**
35	**118**	**39**	**114**	**43**	**110**	**47**	**106**	**51**
102	**55**	**98**	**65**	**88**	**69**	**84**	**73**	**80**
74	**76**	**70**	**72**	**66**	**68**	**56**	**67**	**34**
57	**102**	**53**	**49**	**110**	**45**	**114**	**41**	**118**
37	**34**	**97**	**33**	**91**	**38**	**87**	**42**	**83**
46	**79**	**50**	**75**	**54**	**71**	**69**	**68**	**5**
53	**76**	**49**	**80**	**84**	**41**	**98**	**33**	**100**
103	**106**	**105**	**110**	**111**	**114**	**117**	**118**	**120**

vi) Now convert the values of Table 4 according to the ASCII code value. This matrix will be used for encryption purpose.

Table 5. Encryption Table

z	q	j	%	C	_	X	H	x
#	v	‘	r	+	n	/	j	3
f	7	b	A	X	E	T	I	P
J	L	F	H	B	D	8	C	“
9	f	5	0	N	_	r	)	v
%	“	a	!	[	&	w	*	S
.	O	2	K	6	G	E	D	9
5	L	1	P	T	)	b	!	d
g	j	i	n	o	r	u	v	x

Now encrypt the message **kkgolaa1503@gmail.com.**
*First break the message into groups of two.
kx kg ol ax a1 50 3@ gm ai l. co mx
kx encrypt by **!r**,
kg encrypt by **Pb**,
ol encrypt by **bH,**
ax encrypt by **Ho ,**
a1 encrypt by **HX,**
50 encrypt by **8b,**
3@ encrypt by **l”,**
gm encrypt by **Pj ,**
ai encrypt by **_X,**
l. encrypt by **CT,**
co encrypt by **Dd,**
mx encrypt by **Hn**

1.1 Key Encryption Process

First convert the each character of key MyEmailid into its equivalent ASCII value.
77 121 69 109 97 105 108 105 100 32
Now convert the value into its equivalent binary number of 7 bits.
77: 1001101
121: 1111001
69: 1000101
109: 1101101
97: 1100001
105: 1101001
108: 1101100
105: 1101001
100: 1100100
32: 0100000
Convert the value into equivalent gray code.
1001101 : 1101011
1111001 : 1000101
1000101 : 1100111
1101101 : 1011011
1100001 : 1010001
1101001 : 1011101
1101100 : 1011010
1101001 : 1011101
1100100 : 1010110
0100000 : 0110000
Now convert the gray code into its equivalent integer value.
1101011: 107
1000101: 69
1100111: 103
1011011: 91
1010001: 81
1011101: 93
1011010: 90
1011101: 93
1010110: 86
0110000: 48
Finally convert the values into respective ASCII code which produce the final encrypted key for transmission.
kEg[Q]Z]V0

V. RESULTS AND COMPARISIONS

Table 6 shows the comparison of proposed algorithm and baseline algorithm. The comparison has been with the baseline paper [12]. The proposed work consists the rectangular matrix which includes the alphabets, integer value as well as operators but in [12] the authors have used a matrix which include alphabet in lower case and integer value only. In [12] the size of matrix is 6*6 and the key is encrypted using Caesar cipher technique. A rectangular matrix is used to encrypt the message in the baseline algorithm. In proposed work, alphabets are arranged according to the precedence table. A substitution table is created in the proposed work to provide the more security as compared to baseline algorithm [12]. The proposed algorithm is much secure as compared to baseline algorithm [12].

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Table 6. Comparison with Baseline Algorithm [12]

Parameters	Proposed Algorithm	Baseline Algorithm [12]
Size of Matrix	9*9	6*6
Input Parameters	Alphabets in lower and upper case both, integers (0-9) and list of operators defined in Table2.	Alphabets in lower case and integers (0-9) only
Substitution matrix and ASCII	Substitution matrix and ASCII are used to provide the more security	Not available
Key Encryption	Key encryption is performed using Caesar cipher where the value of n is generated using define method in the proposed algorithm	Key encryption is performed using Caesar cipher where the value of n is taken randomly.

VI. CONCLUSIONS

The conclusion of the research work is that the use of Playfair technique is very useful. The requirement of the security can be done by using this technique. The encryption of key is very useful for more security. This work uses the Substitution matrix and arrange the values according to the ASCII code. It is not necessary that the text will contain only alphabets or integer value, it may be contain alphabets in uppercase or in lower case and many more. So this work generated 9*9 matrix which contain alphabets in uppercase or in lower case both, integer value as well as special symbols. No one can easily detect it even if generates the matrix but would not find the encrypted data till don't know about the rules.

REFERENCES

1. N.A.Putri,A.P.U.Siahaan. F.Wadly and Muslim "Image Similarly Test Using Eigen face Calculation," Int.J.Sci.Technol.,vol.3,no.6,pp.510-515,2018.
2. A.P.U.Siahaan, "High Complexity Bit-Plane Security Enhancement in BPCS Steganography," Int.J.Comput.appl,vol.148,no.3,pp.17-22,2018.
3. A.P.U.Siahaan, "Pelanggaran Cybercrime dankekuatanYuridikshi di Indonesia,"J.Tek.Daninformvol.5, no.1, pp.6-9, 2018.
4. Hariyanto,A.P.U.Siahaan,R.Rahim and Mesran, "Internet protocol security as the network CryptographySystem," Int. J. Sci. Res.Sci.Technol.,vol.3,no.6,pp.223-226,2018.
5. Hariyanto and A.P.U.Siahaan, "Intrusion detection system in Network forensic analysis and," IOSR J.Comput. Sci.Enginee,vol.18,no.6,pp.115-121,2018.
6. Suherman and A.P.U.Siahaan, "Huffman Text Compression Technique," Int. J. Comput. Sci.Enginee ring, vol.3,no.8,pp.103-108,2018.
7. L.Marlina, A.P.U.Siahaan, H.Kurniawan, and I.Sulistianingsih, "Data Compression Using Elias Delta Code," Int. J. Recent Trends Eng. Res., vol.3,no.8,pp.210-217,Aug.2018.
8. M.D.L.Siahaan, M.S.Panjaitan, and A.P.U. Siahaan, "MikroTik Bandwidth Management to Gain the Users Prosperity Prevalent," Int.J.Eng.Trends Technol.,vol.42,no.5,pp.218-222,2018.
9. A.Lubis and A.P.U.Siahaan, "Network Forensic Application in General Cases" IOSR.J.Comput.Eng.,vol.18,no.6,pp.41-44,2018.
10. Mesran Mesran,Imam Solihin, "Implementation of Super Playfair in Messaging," University Pembangunan PancaBudi,Medan,Indonesia.
11. Nisarga Chand and subhajit Bhattacharya "A novel approach for Encryption of Text messages Using PLAYFAIR Cipher 6*6 Matrix with four Iteration Steps", International Journal of Engineering Science and Innovative Technology , vol 3,Issue 1, Jan 2014.
12. Zubair Iqbal, Bhumika Gupta, Kamal Kumar Gola and Prachigupta, "Enhanced the Security of Playfair Technique using Excess 3 code (XS3) and Ceaser Cipher",IJCA (0975-8887)vol 103,no 13, October 2014.
13. Surendra Singh Chauhan,Hawa Singh and Ram NiwasGurjar, " Secure Key Exchange using RSA in extended Playfair Cipher Technique", International journal of Computer Applications (0975-8887) vol 104,no-15, Oct 2014.
14. Kamal Kumar Gola , Zubair Iqbal and Bhumika Gupta, " Modified RSA digital signature scheme for data confidentially", IJCA(0975-8887), vol 106, no 13, Nov 2014.
15. Kamal Kumar Gola, Zubair Iqbal and Bhumika Gupta, " Dual Level Security for key Exchange using Modified RSA Public Key Encryption in Playfair Technique", IJCA(0975-8887) vol 106, no 13, Nov 2014.
16. P.Murli and G. Senthilkumar, and J. Palchodhury, "A Framework for the Development of a New Approach of Playfair Cipher", in Porceedings of India Com 2008, pages 1 -2, Feb 2008.

AUTHORS PROFILE

Kamal Kumar Gola is working as Assistant Professor in Faculty of Engineering, Teerthanker Mahaveer University, Moradabad. He received his B.Tech. Degree from Moradabad Institute of Technology in Computer Science and Engineering and M.Tech. Degree from Uttarakhand Technical University in Computer Science and Engineering. His main research interests are Wireless Sensor Networks, Algorithms and Security.

Manish Dhingra is an engineer and management professional with more than 18 years of experience in industry and academics. He has done BE (Production) from Nagpur University in 1996, Master in Business Administration degree from Kurukshetra University in the year 2003 and secured M.Tech. (Manufacturing Systems) degree in 2015. He started his career as an engineer and served into many companies of International repute and then switched to academics after obtaining his MBA. Presently, he is working as Associate Professor in Faculty of Engineering, Teerthanker Mahaveer University, Moradabad.

Rahul Rathore is M.Tech from Uttarakhand Technical University in Computer Science and Engineering and B.Tech from College of Engineering and Technology, IFTM in Information Technology. He is having 10 years of University teaching experience. He has published more than 15 research papers in international/ national publications. Presently, he is working as Assistant Professor in in Faculty of Engineering, Teerthanker Mahaveer University, Moradabad.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)